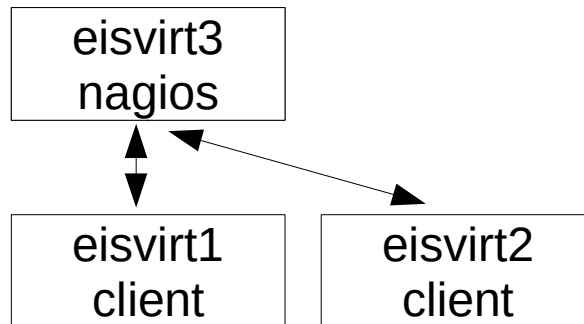


nagios-objects / nrpe Eventhandler

Mit den neuesten Versionen der Pakete nagios-objects und nrpe besteht jetzt die Möglichkeit über die Eisfair-1 Konfigurationsschicht Eventhandler zu konfigurieren.

Hier eine Einführung in die Konfiguration und einige Hinweise zum Test von Eventhandlern.

Es wird folgende Beispielumgebung genutzt:



Host: eisvirt3

Auf dem Host eisvirt3 sind die Pakete nagios, nagios-plugins, pnp4nagios, nagios-objects und nrpe-check installiert. Außerdem natürlich das Paket apache2. Als Beispielobjekt für Local Eventmanager ist zusätzlich pure-ftpd installiert.

Host: eisvirt1

Auf dem Host eisvirt1 sind die Pakete nrpe, nagios-plugins, pure-ftpd und apache2 installiert.

Host: eisvirt2

Hier, auf eisvirt2, sind ebenfalls nrpe, nagios-plugins und pure-ftpd installiert. Als Webserver aber nginx.

Auf allen Hosts sind weitere Pakete verfügbar, die aber bei diesen Beispielen keine Rolle spielen.

Local Eventhandler

Local Eventhandler dienen dazu auf dem lokalen Host, hier also dem Host auf dem Nagios läuft, Services neu zu starten, wenn Nagios feststellt, dass sie nicht mehr funktionieren.

Im Paket nagios-objects ist eine Default-Eventhandler mit dem Namen `local_service_restarter` enthalten. Das Skript steht im Verzeichnis `/usr/local/nagios/eventhandlers`.

Das Skript versucht im Fehlerfall über das Skript `/etc/init.d/<service> restart` den ausgefallenen Service neu zu starten.

Beispiel:
`/etc/init.d/pure-ftpd restart`

Das ist normalerweise nur dem User root erlaubt. Da die Eventhandler und damit das Skript `local_service_restarter` aber unter dem User nagios ausgeführt wird, muss über eine entsprechende sudoers Datei dem User nagios für diese Fälle besondere Rechte gegeben werden.

Beispiel (Datei `/etc/sudoers.d/nagios_local_eventhandler`):

```
#-----  
# /etc/sudoers.d/nagios_local_eventhandler generated by /var/install/config.d/nagios-  
eventhandler.sh  
#  
# Do not edit this file, edit /etc/config.d/nagios-eventhandler  
# Creation date: 2018-01-17 17:16:38  
#-----  
Cmnd_Alias RESTART_APACHE2 = /etc/init.d/apache2 restart  
Cmnd_Alias RESTART_PUREFTPD = /etc/init.d/pure-ftpd restart  
  
nagios ALL=NOPASSWD: RESTART_APACHE2  
nagios ALL=NOPASSWD: RESTART_PUREFTPD
```

Korrektweise ruft also das Skript `local_service_restarter`
`sudo /etc/init.d/<service> restart`
auf.

Um Local Eventhandler für nagios zu konfigurieren wird das Menü
Configure Eventhandler
im Untermenü
Nagios Configuration (Eisfair style)
aufgerufen.

Es stehen folgende Konfigurationsvariablen zur Verfügung:

```
NAGIOS_EVENTHANDLER_N  
NAGIOS_EVENTHANDLER_#_NAME  
NAGIOS_EVENTHANDLER_#_ACTIVE  
NAGIOS_EVENTHANDLER_#_COMMAND  
NAGIOS_EVENTHANDLER_#_SERVICE
```

NAGIOS_EVENTHANDLER_N
Anzahl der Eventhandler (Local und Remote)

NAGIOS_EVENTHANDLER_#_NAME
Name oder Beschreibung des Eventhandlers

NAGIOS_EVENTHANDLER_#_ACTIVE
Ist der Eventhandler aktiv (yes/no)

NAGIOS_EVENTHANDLER_#_COMMAND
Command oder Shell-Skript im Verzeichnis /usr/local/nagios/eventhandlers

NAGIOS_EVENTHANDLER_#_SERVICE
Parameter/ServiceName für das Command bzw. das Shell-Skript.

Die Nutzung des Default-Eventhandlers local_service_restarter würde z.B. über folgende Konfiguration erreicht:

```
NAGIOS_EVENTHANDLER_2_NAME='restart_pureftpd'  
NAGIOS_EVENTHANDLER_2_ACTIVE='yes'  
NAGIOS_EVENTHANDLER_2_COMMAND='local_service_restarter'  
NAGIOS_EVENTHANDLER_2_SERVICE='pure-ftpd'
```

Aus diesen Angaben wird automatisch ein entsprechender Eintrag in der Datei /etc/sudoers.d/nagios_local_eventhandler erzeugt:

```
Cmnd_Alias RESTART_PUREFTPD = /etc/init.d/pure-ftpd restart
```

```
nagios ALL=NOPASSWD: RESTART_PUREFTPD
```

Service mit Eventhandler

Um einen Service im Fehlerfall über den konfigurierten Eventhandler neu zu starten muss die entsprechende Service Konfiguration erweitert werden.

Beispiel:

```
NAGIOS_SERVICE_11_USE='service-interval-5min'
NAGIOS_SERVICE_11_SERVICE_DESCRIPTION='FTP'
NAGIOS_SERVICE_11_HOSTGROUP_N='0'
NAGIOS_SERVICE_11_HOST_N='1'
NAGIOS_SERVICE_11_HOST_1_NAME='localhost'
NAGIOS_SERVICE_11_CHECK_COMMAND='check_ftp'
NAGIOS_SERVICE_11_CHECK_OPTION=""
NAGIOS_SERVICE_11_MAX_CHECK_ATTEMPTS='1'      # Maximum check attempts
NAGIOS_SERVICE_11_EVENT_HANDLER='restart_pureftpd' # Event handler
```

Die Variablen NAGIOS_SERVICE_#_MAX_CHECK_ATTEMPTS und NAGIOS_SERVICE_#_EVENT_HANDLER sind neu!

Bei NAGIOS_SERVICE_11_EVENT_HANDLER wurde der oben konfigurierte Eventhandler angegeben.

NAGIOS_SERVICE_#_MAX_CHECK_ATTEMPTS muss später näher erläutert werden.

Aus dieser Konfiguration generiert das Paket nagios-objects folgende Service Konfiguration:

```
define service{
    use                service-interval-5min,service-pnp
    service_description FTP
    host_name          localhost
    check_command      check_ftp
    max_check_attempts 1
    event_handler      restart_pureftpd
}
```

und folgende Eventhandler Konfiguration:

```
define command{
    command_name      restart_pureftpd
    command_line
/usr/local/nagios/eventhandlers/local_service_restarter $SERVI
CESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$ pure-ftp
}
```

Achtung: die drei Zeilen command line ff. sind hier umgebrochen, es ist nur eine Zeile.

An das Skript

`local_service_restarter`

werden vier Parameter übergeben.

Dies sind die Standard Nagios Service Macros

`$$SERVICESTATES$`

`$$SERVICESTATETYPE$`

`$$SERVICEATTEMPT$`

und zusätzlich

`NAGIOS_EVENTHANDLER_1_SERVICE`

hier also der Wert `pure-ftpd`.

`$$SERVICESTATES$` hat die Werte OK, WARNING, UNKNOWN und CRITICAL

`$$SERVICESTATETYPE$` hat die Werte SOFT und HARD

`$$SERVICEATTEMPT$` ist ein Zähler von 1 bis `max_check_attempts`

Das Skript `local_service_restarter` reagiert nur auf den Wert CRITICAL von `$$SERVICESTATES$`.

Nimmt `$$SERVICESTATETYPE$` den Wert SOFT an, so reagiert das Skript erst wenn `$$SERVICEATTEMPT$` den Wert 3 hat.

Das Skript reagiert sofort, wenn `$$SERVICESTATETYPE$` den Wert HARD hat.

Bei `max_check_attempts` 4 wird das Skript von nagios im Fehlerfall viermal aufgerufen:

```
CRITICAL SOFT 1
CRITICAL SOFT 2
CRITICAL SOFT 3
CRITICAL HARD 4
```

Da erst bei SOFT und 3 reagiert wird, kann je nach Konfiguration des Service Intervalls eine längere Zeit vergehen, bis reagiert wird. Bei `service-interval-5min` also bis zu 9 Minuten.

Hinweis:

```
define service{
    name                service-interval-5min
    use                 master-service-template
    check_interval      5
    retry_check_interval 2
    register            0
}
```

`retry_check_interval` steht auf 2

Wird `max_check_attempts` auf 1 gesetzt, so ist schon der erste Aufruf im Fehlerfall

```
CRITICAL HARD 1
```

Dann reagiert das Skript also beim ersten Auftreten eines Fehlers.

Wird zusätzlich `service-interval-1min` gesetzt, so erfolgt die Reaktion bereits nach spätestens 2 Minuten.

Hier gilt wie immer: je kürzer das Service Intervall desto öfter erfolgt der Check und desto schneller kann reagiert werden, aber desto höher ist auch die Belastung des Nagios Hosts und natürlich des

überwachten Clients.

Testhilfen:

Das Skript `local_service_restarter` bietet die Möglichkeit ein Logging der Aufrufe einzuschalten.

```
....  
# set LOGGING to 'yes' to turn on logging to /tmp/local_service_restarter.log  
LOGGING='yes'  
....
```

In der Datei `/tmp/local_service_restarter.log` werden dann die Aufrufe notiert.

Beispiel:

```
local_service_restarter: Mon Jan 29 14:06:32 CET 2018 Parameter: CRITICAL HARD 1  
pure-ftpd  
Restarting service pure-ftpd...  
  * Stopping Pure-FTPd daemon ...      Not running.                [ WARN ]  
  * Starting Pure-FTPd daemon ...      [ OK ]  
local_service_restarter: Mon Jan 29 14:11:32 CET 2018 Parameter: OK HARD 1 pure-  
ftpd
```

Hier wurde gemäß der obigen Konfiguration

```
max_check_attempts 1
```

bereits beim ersten Aufruf die Parameter

```
CRITICAL HARD 1 pure-ftpd
```

übergeben.

Offensichtlich war der Restart des `pure-ftpd` erfolgreich. Dies sieht man im Protokoll und in der nächsten Zeile, 5 Minute später:

```
OK HARD 1 pure-ftpd
```

Ausgabe bei `service-interval-5min` und `max_check_attempts 4` bei einem überwachten `apache2`:

```
local_service_restarter: Fri Jan 26 16:35:13 CET 2018 Parameter: CRITICAL SOFT 1  
apache2  
local_service_restarter: Fri Jan 26 16:37:13 CET 2018 Parameter: CRITICAL SOFT 2  
apache2  
local_service_restarter: Fri Jan 26 16:39:13 CET 2018 Parameter: CRITICAL SOFT 3  
apache2  
Restarting service apache2 (3rd soft critical state)...  
  * Restarting Apache ...  
httpd not running, trying to start  
httpd not running, trying to start  
=> OK  
local_service_restarter: Fri Jan 26 16:41:13 CET 2018 Parameter: OK SOFT 4  
apache2
```

Es ist auch immer hilfreich das Log von `nagios` im Auge zu behalten.

Hier ein kleiner Trick, um das Dateiformat lesbarer zu machen.

Mittels

```
tail -f /var/lib/nagios/nagios.log | perl -pe 's/(\d+)/localtime($1)/e'
```

lassen sich die Einträge mit lesbaren Datums/Uhrzeitangaben ausgeben. Ende mit `CTRL-C`.

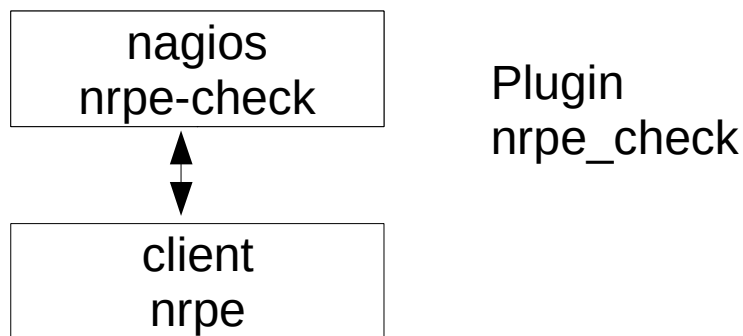
Remote Eventhandler

Remote Eventhandler dienen dazu Services auf einem remote Host neu zu starten. Dazu ist natürlich eine Kommunikation zwischen dem Nagios Host und dem remote Host vorhanden sein.

Als Lösungsweg wurden hier Nagios Remote Process Execution mit den Paketen nrpe_check und nrpe gewählt.

Das Paket nrpe_check wird oder ist bereits auf dem Nagios Host installiert und stellt das sog. Plugins nrpe_check zur Verfügung. Das paket nrpe wird oder ist bereits auf dem remote Host installiert und bietet über den Service nrpe die Möglichkeit Kommandos auf dem remote Host ausführen zu lassen.

Oftmals dienen die Pakete nrpe-check und nrpe bereits zur Überwachung der remote Host, die Remote Eventhandler wurden daher in das Paket nrpe integriert.



Im Paket nagios-objects ist eine Default-Remote-Eventhandler mit dem Namen remote_service_restarter enthalten. Das Skript steht im Verzeichnis /usr/local/nagios/eventhandlers.

Hier werden erneut die folgenden Konfigurationsvariablen genutzt:

```
NAGIOS_EVENTHANDLER_N
NAGIOS_EVENTHANDLER_#_NAME
NAGIOS_EVENTHANDLER_#_ACTIVE
NAGIOS_EVENTHANDLER_#_COMMAND
NAGIOS_EVENTHANDLER_#_SERVICE
```

Beispiel:

```
NAGIOS_EVENTHANDLER_5_NAME='remote_restart_httpd'
NAGIOS_EVENTHANDLER_5_ACTIVE='yes'
NAGIOS_EVENTHANDLER_5_COMMAND='remote_service_restarter'
NAGIOS_EVENTHANDLER_5_SERVICE='restart_httpd'
```

Um mit dem remote Host kommunizieren zu können müssen Informationen über den Host, der zu nutzende Port vorhanden sein. Zusätzlich muss bekannt sein, ob verschlüsselt oder unverschlüsselt kommuniziert wird.

Um diese Information bereits zu stellen gibt es zwei Möglichkeiten:

- a.) der remote Host ist bereits in der Konfiguration des Paketes nrpe enthalten
- b.) die Angaben müssen für den Eventhandler bereit gestellt werden.

In nagios-objects, Datei /etc/config.d/nagios-eventhandler existieren dazu folgende Konfigurationsvariablen:

```
NAGIOS_RMHOST_INFO_N
NAGIOS_RMHOST_INFO_#_HOSTNAME
NAGIOS_RMHOST_INFO_#_ACTIVE
NAGIOS_RMHOST_INFO_#_CONFIG
```

```
NAGIOS_RMHOST_INFO_N
Anzahl der Informationen für remote Hosts
NAGIOS_RMHOST_INFO_#_HOSTNAME
Hostname
NAGIOS_RMHOST_INFO_#_ACTIVE
Ist die Konfiguration aktiv (yes/no)
NAGIOS_RMHOST_INFO_#_CONFIG
Konfiguration
    nrpe                Informationen für diesen Host finden sich in /etc/config.d/nrpe-check
    <port>:ssl          Es wird der Port <port> mit Verschlüsselung genutzt
    <port>:no_ssl       Es wird der Port <port> ohne Verschlüsselung genutzt
```

Sind Informationen zu einem Host bereits in /etc/config.d/nrpe-check sollte unbedingt 'nrpe' genutzt werden, da sonst die Informationen doppelt gehalten werden.

Beispiel:

```
NAGIOS_RMHOST_INFO_N='2'
NAGIOS_RMHOST_INFO_1_HOSTNAME='eisvirt1'
NAGIOS_RMHOST_INFO_1_ACTIVE='yes'
NAGIOS_RMHOST_INFO_1_CONFIG='nrpe'
NAGIOS_RMHOST_INFO_2_HOSTNAME='eisvirt2'
NAGIOS_RMHOST_INFO_2_ACTIVE='yes'
NAGIOS_RMHOST_INFO_2_CONFIG='5667:no_ssl'
```

Hier wurde testweise für den Host 'eisvirt2' die Information direkt angegeben.

Die Remote Host Konfiguration aus /etc/config.d/nagios-eventhandler "landet" in der Datei /etc/nagios/restarter.conf.

Beispiel:

```
#-----
# /etc/nagios/restarter.conf generated by /var/install/config.d/nagios-
eventhandler.sh
#
# Do not edit this file, edit /etc/config.d/nagios-eventhandler
# Creation date: 2018-01-17 17:16:38
#-----
eisvirt1:192.168.178.71:5666:ssl
eisvirt2:192.168.178.72:5667:no_ssl
```


Um einen Remote Service im Fehlerfall über den konfigurierten Eventhandler neu zu starten muss die entsprechende Service Konfiguration erweitert werden.

Beispiel:

```
NAGIOS_SERVICE_9_USE='service-interval-1min'  
NAGIOS_SERVICE_9_SERVICE_DESCRIPTION='HTTP'  
NAGIOS_SERVICE_9_HOSTGROUP_N='0'  
NAGIOS_SERVICE_9_HOST_N='2'  
NAGIOS_SERVICE_9_HOST_1_NAME='eisvirt1'  
NAGIOS_SERVICE_9_HOST_2_NAME='eisvirt2'  
NAGIOS_SERVICE_9_CHECK_COMMAND='check_http'  
NAGIOS_SERVICE_9_CHECK_OPTION=''  
NAGIOS_SERVICE_9_MAX_CHECK_ATTEMPTS='1'  
NAGIOS_SERVICE_9_EVENT_HANDLER='remote_restart_httpd'
```

```
NAGIOS_EVENTHANDLER_5_NAME='remote_restart_httpd'  
NAGIOS_EVENTHANDLER_5_ACTIVE='yes'  
NAGIOS_EVENTHANDLER_5_COMMAND='remote_service_restarter'  
NAGIOS_EVENTHANDLER_5_SERVICE='restart_httpd'
```

Hier werden gleich zwei Remote Hosts durch den Check 'check_http' überwacht. Beide nutzen den gleichen Eventhandler 'remote_restart_httpd' und die gleiche NAGIOS_EVENTHANDLER_#_SERVICE Angabe 'restart_httpd'.

Aus diesen Angaben resultieren die folgende Service Konfiguration:

```
define service{  
    use                service-interval-1min,service-pnp  
    service_description HTTP  
    host_name          eisvirt1,eisvirt2  
    check_command      check_http  
    max_check_attempts 1  
    event_handler      remote_restart_httpd  
}
```

und folgende Eventhandler Konfiguration:

```
define command{  
    command_name      remote_restart_httpd  
    command_line  
/usr/local/nagios/eventhandlers/remote_service_restarter  
$SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$ $HOSTNAME$  
restart_httpd  
}
```

Achtung: die drei Zeilen command line ff. sind hier umgebrochen, es ist nur eine Zeile.

An das Skript

```
remote_service_restarter
```

werden fünf Parameter übergeben.

Dies sind die Standard Nagios Service Macros

`$SERVICESTATES`
`$SERVICESTATETYPES`
`$SERVICEATTEMPT$`
`$HOSTNAMES`

und zusätzlich
`NAGIOS_EVENTHANDLER_5_SERVICE`
hier also der Wert `restart_http`.

Das Skript `remote_service_restarter` reagiert bezüglich `$SERVICESTATES`,
`$SERVICESTATETYPES` und `$SERVICEATTEMPT$` wie `local_service_restarter`.

Um im Fehlerfall den Service auf dem Remote Host neu zu starten kommt jetzt das Plugin
`nrpe_check` und die Konfiguration von `nrpe` auf dem Remote Host (siehe unten) ins Spiel.

Im Fehlerfall wird das Plugin `nrpe:check` wie folgt aufgerufen:

eisvirt1:
`/usr/local/nagios/plugins/check_nrpe -H 192.168.178.71 -p 5666 -c restart_httpd`

eisvirt2:
`/usr/local/nagios/plugins/check_nrpe -n -H 192.168.178.72 -p 5667 -c restart_httpd`

Hier wird deutlich warum Angaben zur IP des Hosts, dem zu nutzenden Port und zur
Verschlüsselung (Aufruf mit/ohne Parameter `-n`) notwendig sind. Es wäre auch möglich gewesen
statt `$HOSTNAMES` an dieser Stelle direkt `$HOSTADDESS$` zu nutzen. Dies wurde aber
verworfen, um möglichst durchgehend mit Hostnamen zu arbeiten.

Man beachte bitte, dass auf `eisvirt1` eine `Apache2` als Webserver genutzt wird, auf `eisvirt2` aber
`nginx`.

nrpe Konfiguration

Die remote Host Konfiguration für Eventhandler wurde in die Konfiguration des Paketes nrpe integriert.

Die Konfiguration in der Datei /etc/config.d/nrpe wurde um folgende Variablen erweitert:

```
NRPE_EVENTHANDLER_N
NRPE_EVENTHANDLER_#_ACTIVE
NRPE_EVENTHANDLER_#_NAME
NRPE_EVENTHANDLER_#_COMMAND
NRPE_EVENTHANDLER_#_SERVICE
```

NRPE_EVENTHANDLER_N
Anzahl der Eventhandler

NRPE_EVENTHANDLER_#_ACTIVE
Soll der Eventhandler aktiv sein yes|no

NRPE_EVENTHANDLER_#_NAME
Name des Eventhandlers z.B. restart_<service>
Darf nicht mit check beginnen

NRPE_EVENTHANDLER_#_COMMAND
Shell Skript im Verzeichnis /usr/local/nagios/eventhandlers

NRPE_EVENTHANDLER_#_SERVICE
Parameter/ServiceName für NRPE_EVENTHANDLER_#_COMMAND
Bei Nutzung von /usr/local/nagios/eventhandlers/service_restarter
wird ein 'sudo /etc/init.d/<service> restart' ausgeführt

Als remote Teil für die Ausführung von Restart Befehlen dient das Skript
service_restarter
Das Skript steht im Verzeichnis /usr/local/nagios/eventhandlers.

Da auch hier wieder ein sudo notwendig ist wird bei Nutzung des Skript service_restarter automatisch ein entsprechender Eintrag in der Datei /etc/sudoers.d/nrpe erzeugt.

Beispiel:

```
#-----
# /etc/sudoers.d/nrpe generated by /var/install/config.d/nrpe.sh
#
# Do not edit this file, edit /etc/config.d/nrpe
# Creation date: 2018-01-14 16:56:43
#-----
Cmnd_Alias RESTART_PUREFTPD = /etc/init.d/pure-ftpd restart
Cmnd_Alias RESTART_HTTPD = /etc/init.d/apache2 restart

nagios ALL=NOPASSWD: RESTART_PUREFTPD
nagios ALL=NOPASSWD: RESTART_HTTPD
```

Beispielkonfiguration auf eisvirt1:

```
NRPE_EVENTHANDLER_3_ACTIVE='yes'  
NRPE_EVENTHANDLER_3_NAME='restart_httpd'  
NRPE_EVENTHANDLER_3_COMMAND='service_restarter'  
NRPE_EVENTHANDLER_3_SERVICE='apache2'
```

Beispielkonfiguration auf eisvirt2:

```
NRPE_EVENTHANDLER_1_ACTIVE='yes'  
NRPE_EVENTHANDLER_1_NAME='restart_httpd'  
NRPE_EVENTHANDLER_1_COMMAND='service_restarter'  
NRPE_EVENTHANDLER_1_SERVICE='nginx'
```

Testhilfen:

Auch das Skript `remote_service_restarter` auf dem Nagios Host bietet die Möglichkeit ein Logging der Aufrufe einzuschalten.

```
....  
# set LOGGING to 'yes' to turn on logging to /tmp/remote_service_restarter.log  
LOGGING='no'  
....
```

In der Datei `/tmp/remote_service_restarter.log` werden die Aufrufe notiert:

```
remote_service_restarter: Sun Jan 28 11:01:53 CET 2018 Parameter: CRITICAL HARD  
1 eisvirt2 restart_httpd  
Restarting restart_httpd with nrpe command...  
/usr/local/nagios/plugins/check_nrpe -n -H 192.168.178.72 -p 5667 -c  
restart_httpd
```

Auch der `service_restarter` auf dem remote Host bietet eine Logging Möglichkeit.

```
....  
# set LOGGING to 'yes' to turn on logging to /tmp/service_restarter.log  
LOGGING='no'  
....
```

Beispiel:

```
service_restarter: Thu Jan 18 13:41:27 CET 2018 Service: nginx  
Restarting nginx service...  
* Stopping nginx daemon ... Not running.  
[ WARN ]  
* Starting nginx daemon ...  
[ OK ]  
OK
```

Hier noch ein Hinweis, wie man testen kann, ob die generierte sudo Datei korrekt ist.

1.) Test

Beispiel:

```
NRPE_EVENTHANDLER_2_ACTIVE='yes'  
NRPE_EVENTHANDLER_2_NAME='restart_pureftpd'  
NRPE_EVENTHANDLER_2_COMMAND='service_restarter'  
NRPE_EVENTHANDLER_2_SERVICE='pure-ftp'
```

```
su nagios -s /bin/sh -c 'sh /usr/local/nagios/eventhandlers/service_restarter pure-ftp'
```

OK

Kontrolle beim pure-ftp
/etc/init.d/pure-ftp status

Achtung:

Nicht alle Services ändern beim restart die PID (z.B. apache2)

Bei Problemen: Datei /etc/sudoers.d/nrpe kontrollieren.

2.) Negativtest

Es gibt keinen Eintrag mit NRPE_EVENTHANDLER_#_SERVICE='gehtnicht'

```
su nagios -s /bin/sh -c 'sh /usr/local/nagios/eventhandlers/service_restarter gehtnicht'
```

CRITICAL: sudo returned 1

3.) Negativtest

```
su nagios -s /bin/sh -c 'sh /usr/local/nagios/eventhandlers/service_restarter'
```

CRITICAL: service name missing