



## **Einbindung des Kentix AlarmManager-PRO in Nagios**

25. März 2013  
B1 Systems GmbH

## Inhaltsverzeichnis

<b>1</b>	<b>Überwachung des AlarmManager-PRO</b>	<b>2</b>
1.1	Konfiguration/Einbinden in Nagios . . . . .	3
1.2	check_KAMP.pl --help . . . . .	4
<b>2</b>	<b>Überwachung der AlarmManager-PRO Sensoren</b>	<b>6</b>
2.1	Konfiguration/Einbinden in Nagios . . . . .	7
2.2	check_KAMP_sensors.pl --help . . . . .	9
<b>3</b>	<b>SMS versenden über AlarmManager-PRO</b>	<b>11</b>
3.1	Konfiguration/Einbinden in Nagios . . . . .	11
3.2	send_sms_AlarmManagerPro.pl --help . . . . .	12
<b>4</b>	<b>B1 Systems</b>	<b>13</b>
4.1	Support . . . . .	13
4.2	Consulting . . . . .	13

## 1 Überwachung des AlarmManager-PRO

Mit Hilfe des Plugins `check_KAMP` ist es möglich, einen AlarmManager-PRO der Kentix GmbH mit Nagios zu überwachen. Die Daten werden mit Hilfe des Simple Network Management Protocol (SNMP) abgefragt.

Folgende Funktionen können abgefragt werden:

- Scharfschaltung der Alarmzonen 1-3
- Zustand der Alarmer 1 und 2
- Zustand der Serverabfragen
- Zustand der Sensorenkommunikation
- Status des externen Alarms
- Status der externen Stromversorgung
- Zustand des Sabotagealarms
- Stärke des GSM-Signals
- Status des GSM-Moduls

Listing 1: Beispielausgabe der Statusabfrage des GSM-Moduls

```
$> ./check_KAMP.pl --extra-opts=check_KAMP -m gsm  
check_KAMP OK - gsmok: 1, gsmsignal: 3 | gsmsignal=3;;
```

## 1.1 Konfiguration/Einbinden in Nagios

Es empfiehlt sich, den Communitystring für die Datenabfrage per SNMP zentral zu hinterlegen. Hier gibt es zwei Möglichkeiten, zum einem in der `resource.cfg` in `$(USERx$)` Macros, zum anderen in der `plugins.ini`-Datei. In der `plugins.ini`-Datei können Parameter für Plugins hinterlegt werden, welche den Parameter `--extra-opts1` unterstützen.

Listing 2: `/etc/nagios/resource.cfg`

```
# Sets $(USER1$) to be the path to the plugins
$(USER1$)=/usr/lib/nagios/plugins

$(USER2$)=snmpreadcommunitystring
```

Eine dazu passende Command- und Servicedefinition:

Listing 3: Eintrag in z.B. `/etc/nagios/objects/KAMP.cfg`

```
define command{
    command_name check_KAMP
    command_line $(USER1$)/check_KAMP.pl -H $HOSTADDRESS$ -C $(USER2$) -m
                $ARG1$
}

define service{
    use generic-service
    host_name alarmmanagerpro
    service_description GSM
    check_command check_KAMP!gsm
    normal_check_interval 10
    retry_check_interval 1
}
```

Ein Beispiel für einen Eintrag in der `plugins.ini`-Datei, um den Parameter `--extra-opts` zu nutzen, kann so aussehen:

Listing 4: `plugins.ini`

```
[check_KAMP]
community=snmpreadcommunitystring
```

<sup>1</sup>siehe <http://nagiosplugins.org/extra-opts>

Dazu die Command- und Servicedefinition:

Listing 5: Eintrag in z.B. `/etc/nagios/objects/KAMP.cfg` mit `--extra-opts`

```
define command{
    command_name check_KAMP
    command_line $USER1$/check_KAMP.pl --extra-opts=check_KAMP -H
                $HOSTADDRESS$ -m $ARG1$
}

define service{
    use generic-service
    host_name alarmmanagerpro
    service_description GSM
    check_command check_KAMP!gsm
    normal_check_interval 10
    retry_check_interval 1
}
```

Standardmäßig wird an diesen Stellen nach der `plugins.ini`-Datei gesucht:

- `/etc/nagios/plugins.ini`
- `/usr/local/nagios/etc/plugins.ini`
- `/usr/local/etc/nagios/plugins.ini`
- `/etc/opt/nagios/plugins.ini`
- `/etc/nagios-plugins.ini`
- `/usr/local/etc/nagios-plugins.ini`
- `/etc/opt/nagios-plugins.ini`

## 1.2 `check_KAMP.pl --help`

Die eingebaute Hilfe kann mit `check_KAMP --help` aufgerufen werden. Diese zeigt neben den benutzten Langparametern auch die jeweilige Kurzversion sowie eine kurze Beschreibung an.

Listing 6: Hilfe-Funktion des Plugins

```
$> check_KAMP.pl --help
check_KAMP.pl 1.0.0

GPLv2
```

This Plugin checks a Kentix Alarmmanager-Pro.

```
Usage: check_KAMP.pl [ -v|--verbose ] -H <host> -C <communitystring>
      -m <module>[-t <timeout>] [ -e|--expects=<expected value> ]
```

```
-?, --usage
    Print usage information
-h, --help
    Print detailed help screen
-V, --version
    Print version information
--extra-opts=[section][@file]
    Read options from an ini file. See http://nagiosplugins.org/extra-opts for usage and examples.
-H, --host=STRING
    hostname or IP of Kentix Alarmmanager-Pro
-C, --community=STRING
    community string for snmp connection
-m, --module=STRING
    module to check: alarmzone1, alarmzone2, alarmzone3, alarm1,
    alarm2, serverstate, sensorcommunication, extalarm, extarmed,
    extpower, sabotage, gsm signal, gsmok
-e, --expects=INTEGER
    expected value if checking alarmzone1, alarmzone2, alarmzone3,
    alarm1, alarm2, serverstate, sensorcommunication, extalarm,
    extarmed, extpower or sabotage
-w, --warning=INTEGER
    warning threshold
-c, --critical=INTEGER
    critical threshold
-t, --timeout=INTEGER
    Seconds before plugin times out (default: 15)
-v, --verbose
    Show details for command-line debugging (can repeat up to 3 times)
```

## 2 Überwachung der AlarmManager-PRO Sensoren

Mit Hilfe des Plugins `check_KAMP_sensors` können die Sensoren des AlarmManager-PRO ausgelesen und die Daten an Nagios übergeben werden. Die dabei erhobenen Performancedaten können in Addons wie z.B. PNP4Nagios<sup>2</sup> verarbeitet werden.

Die Daten werden mit Hilfe des Simple Network Management Protocols (SNMP) abgefragt.

Folgende Sensoren können abgefragt werden:

- Temperatur
- Luftfeuchtigkeit
- Taupunkt
- Kohlenmonoxidgehalt

### Listing 7: Abfrage der Temperatur

```
$> ./check_KAMP_sensors.pl --host alarmpro.example.org \  
  --community public --sensor 1 --warning-temperature 23 \  
  --critical-temperature 35 --sensortype temperature  
check_KAMP_sensors WARNING - LAB1, temperature is WARNING,  
  temperature: 23.6 | temperature=23.6;23;35
```

Verzichtet man auf die Angabe des abzufragenden Sensortypen mit dem Parameter `--sensortype`, so werden alle Sensortypen abgefragt:

### Listing 8: Abfrage aller Sensortypen

```
$> ./check_KAMP_sensors.pl --host alarmpro.example.org \  
  --community public --sensor 1 --warning-temperature 23 \  
  --critical-temperature 35  
check_KAMP_sensors WARNING - LAB1, temperature is WARNING,  
  temperature: 23.6, humidity: 31.4, dewpoint: 5.7, co: 0 |  
  temperature=23.6;23;40 humidity=31.4;; dewpoint=5.7;; co=0;;
```

Die Schwellwerte für `warning` und `critical` müssen mit den jeweiligen Parametern separat angegeben werden.

---

<sup>2</sup><http://www.pnp4nagios.org>

## 2.1 Konfiguration/Einbinden in Nagios

Es empfiehlt sich, den Communitystring für die Datenabfrage per SNMP zentral zu hinterlegen. Hier gibt es zwei Möglichkeiten, zum einem in der `resource.cfg` in `$(USERx$)` Macros, zum anderen in der `plugins.ini`-Datei. In der `plugins.ini`-Datei können Parameter für Plugins hinterlegt werden, die den Parameter `--extra-opts`<sup>3</sup> unterstützen.

Listing 9: `/etc/nagios/resource.cfg`

```
# Sets $(USER1$) to be the path to the plugins
$(USER1$)=/usr/lib/nagios/plugins

$(USER2$)=snmpreadcommunitystring
```

Eine dazu passende Command- und Servicedefinition:

Listing 10: Eintrag in z.B. `/etc/nagios/objects/KAMP_sensors.cfg`

```
define command{
    command_name check_KAMP_sensors
    command_line $(USER1$)/check_KAMP_sensors.pl -H $HOSTADDRESS$ -C
        $(USER2$) --sensor $ARG1$ --sensortype $ARG2$ --warning-
        temperature $ARG3$ --critical-temperature $ARG4$
}

define command{
    command_name check_KAMP_sensors_all
    command_line $(USER1$)/check_KAMP_sensors.pl -H $HOSTADDRESS$ -C
        $(USER2$) --sensor $ARG1$
}

define service{
    use generic-service
    host_name alarmmanagerpro
    service_description LAB1 temperature
    check_command check_KAMP_sensors!1!temperature!23!35
    normal_check_interval 10
    retry_check_interval 1
}

define service{
    use generic-service
    host_name alarmmanagerpro
    service_description LAB1 overview
    check_command check_KAMP_sensors_all!1
```

<sup>3</sup>siehe <http://nagiosplugins.org/extra-opts>



```
normal_check_interval 10
retry_check_interval 1
}
```

Ein Beispiel für einen Eintrag in der `plugins.ini`-Datei, um den Parameter `--extra-opts` zu nutzen, kann so aussehen:

Listing 11: Eintrag in `plugins.ini`

```
[check_KAMP]
community=snmpreadcommunitystring
```

Dazu die Command- und Servicedefinition:

Listing 12: Eintrag in z.B. `/etc/nagios/objects/KAMP_sensors.cfg` mit `--extra-opts`

```
define command{
    command_name check_KAMP_sensors
    command_line $USER1$/check_KAMP_sensors.pl -H $HOSTADDRESS$ --
        extra-opts=check_KAMP_sensors --sensor $ARG1$ --sensortype
        $ARG2$ --warning-temperature $ARG3$ --critical-temperature
        $ARG4$
}

define command{
    command_name check_KAMP_sensors_all
    command_line $USER1$/check_KAMP_sensors.pl -H $HOSTADDRESS$ --
        extra-opts=check_KAMP_sensors --sensor $ARG1$
}

define service{
    use generic-service
    host_name alarmmanagerpro
    service_description LAB1 temperature
    check_command check_KAMP_sensors!1!temperature!23!35
    normal_check_interval 10
    retry_check_interval 1
}

define service{
    use generic-service
    host_name alarmmanagerpro
    service_description LAB1 overview
    check_command check_KAMP_sensors!1
    normal_check_interval 10
    retry_check_interval 1
}
```

Standardmäßig wird an diesen Stellen nach der `plugins.ini`-Datei gesucht:

- `/etc/nagios/plugins.ini`
- `/usr/local/nagios/etc/plugins.ini`
- `/usr/local/etc/nagios/plugins.ini`
- `/etc/opt/nagios/plugins.ini`
- `/etc/nagios-plugins.ini`
- `/usr/local/etc/nagios-plugins.ini`
- `/etc/opt/nagios-plugins.ini`

## 2.2 `check_KAMP_sensors.pl --help`

Die eingebaute Hilfe kann mit `check_KAMP_sensors.pl --help` aufgerufen werden. Diese zeigt neben den benutzten Langparametern auch die jeweilige Kurzversion sowie eine kurze Beschreibung an.

Listing 13: Hilfe-Funktion des Plugins

```
check_KAMP_sensors.pl 1.0.0

GPLv2

This Plugin checks the sensors connected to a Kentix AlarmManager Pro
.

Usage: check_KAMP_sensors.pl [ -v|--verbose ] -H <host> -C <
communitystring> [-s <sensor>][-t <timeout>] [ -c|--critical=<
critical threshold> ] [ -w|--warning=<warning threshold> ]

-?, --usage
    Print usage information
-h, --help
    Print detailed help screen
-V, --version
    Print version information
--extra-opts=[section][@file]
    Read options from an ini file. See http://nagiosplugins.org/extra-opts
    for usage and examples.
--host, -H
```

```
hostname or IP of Kentix Alarmmanager Pro
--community, -C
  community string for snmp connection
--sensor, -s
  number of sensor to check, if not set all sensors are checked
--sensortype, -t
  number of sensor to check, if not set all sensors are checked
  valid types: temperature, humidity, dewpoint, co
--warning-temperature
  warning value for temperature
--critical-temperature
  critical value for temperature
--warning-humidity
  warning value for humidity
--critical-humidity
  critical value for humidity
--warning-dewpoint
  warning value for dewpoint
--critical-dewpoint
  critical value for dewpoint
--warningco
  warning value for co
--criticalco
  critical value for co
-t, --timeout=INTEGER
  Seconds before plugin times out (default: 15)
-v, --verbose
  Show details for command-line debugging (can repeat up to 3 times)
```

### 3 SMS versenden über AlarmManager-PRO

Der AlarmManager-PRO bietet die Möglichkeit, mit Hilfe einer Webschnittstelle über das eingebaute GSM-Modem SMS-Nachrichten zu versenden.

Das Plugin `send_sms_AlarmManagerPro.pl` nutzt diese Schnittstelle und kann genutzt werden, um Alarmmeldungen von Nagios per SMS an die Kontakte zu versenden.

Listing 14: Beispielaufruf in der Shell

```
$> ./send_sms_AlarmManagerPro.pl -a alarmpro.example.org \  
--user smsuser --password smspass --recipient 0491234567890 \  
--smstext "MAINDB is DOWN, ping timeout on 2013-01-07"  
OK
```

#### 3.1 Konfiguration/Einbinden in Nagios

Für die Konfiguration des SMS-Skripts bietet sich wieder an, die Informationen wie Serveradresse sowie Benutzer und Passwort zum Versenden der SMS in der Resource-Konfigurationsdatei zu hinterlegen.

Listing 15: `/etc/nagios/resource.cfg`

```
# Sets $USER1$ to be the path to the plugins  
$USER1$=/usr/lib/nagios/plugins  
  
$USER2$=alarmpro.example.org  
$USER3$=smsuser  
$USER4$=smspass
```

Die benötigten Kommandos zum Versenden der Nachrichten können dann wie folgt aussehen:

Listing 16: Eintrag in z.B. `/etc/nagios/objects/send_SMS.cfg`

```
define command{  
    command_name notify-host-by-sms  
    command_line $USER1$/send_sms_AlarmManagerPro.pl -a $USER2$ --  
        user $USER3$ --password $USER4$ --recipient $CONTACTPAGER$ --  
        smstext "$HOSTALIAS$ is $HOSTSTATE$, $HOSTOUTPUT$ at  
        $LONGDATETIME$"  
}  
  
define command{
```

```
command_name notify-service-by-sms
command_line $USER1$/send_sms_AlarmManagerPro.pl -a $USER2$ --
  user $USER3$ --password $USER4$ --recipient $CONTACTPAGER$ --
  smstext "$SERVICEALIAS$ on $HOSTALIAS$ is $SERVICESTATES$,
  $SERVICEDESC at $LOGDATETIME$"
}
```

Die Definitionen der Kontakte, welche per SMS benachrichtigt werden sollen, müssen um das Feld „pager“ erweitert, sowie Kommandos für Host- bzw. Servicebenachrichtigung passend gesetzt werden.

Listing 17: Beispielauszug einer Kontaktdefinition

```
define contact{
  contact_name          john_doe
  host_notification_commands  notify-host-by-sms
  service_notification_commands  notify-service-by-sms
  pager                00491234567890
  ...
}
```

### 3.2 send\_sms\_AlarmManagerPro.pl --help

Mit `send_sms_AlarmManagerPro.pl --help` kann die eingebaute Hilfe angezeigt werden, welche neben den benutzten Langparametern auch die jeweilige Kurzversion sowie eine kurze Beschreibung anbietet.

Listing 18: Ausgabe der Hilfe

```
$> send_sms_AlarmManagerPro.pl --help
Usage: send_sms_AlarmManagerPro.pl --alarmmanagerpro <host> --user <
  user> --password <password> --recipient <phonenummer> --smstext [-
  t <timeout>]

--alarmmanagerpro, -a  Hostname/IP of AlarmManagerPro
--user, -p             name of Kentix AlarmManagerPro user allowed to
  send SMS
--password, -p        password for AlarmManagerPro user allowed to
  send SMS
--recipient, -r       recipient
--smstext, -s         SMS text, will be cut of after 160 chars
--timeout, -t         set timeout in seconds (default: 15)
--help, --usage, -h  print this text
--version, -v         print version
```

## 4 B1 Systems

B1 Systems ist mit seinem Team von mehr als 60 Mitarbeitern auf Bereiche im Linux® und Open Source-Umfeld spezialisiert und hat in der Entwicklung und Implementierung von z.B.:

- Hochverfügbarkeit (Pacemaker)
- Virtualisierung (Xen & KVM)
- System-Management (Spacewalk, SUSE Manager, Red Hat Network Satellite)
- Monitoring (Nagios)
- Cloud (OpenStack, Suse Cloud, ownCloud)

Referenzstatus erreicht. Die B1 Systems bietet ihren Kunden Lösungen aus einer Hand an und begleitet sie in den unterschiedlichen Bereichen wie z.B. Konzeptionierung, Umsetzung, Schulung & Support.

### 4.1 Support

Mit unserem Support decken wir nicht nur die üblichen Geschäftszeiten ab, sondern bieten diesen auch rund um die Uhr an. Unsere qualifizierten Reaktionszeiten bieten wir unseren Kunden über unterschiedliche Schnittstellen an wie z.B. Hotline, Ticket-system, eMail.

### 4.2 Consulting

Mit unseren Architekten, Consultants und Entwicklern erarbeitet B1 Systems in enger Zusammenarbeit mit seinen Kunden Lösungen, die sich reibungslos in bestehende Umgebungen und Betriebsprozesse integrieren lassen. Beispielsweise entwickeln wir maßgeschneiderte Monitoring-Lösungen auf Basis von Nagios und implementiert diese nach Kundenbedürfnissen - individuell und passgenau.