

check_cisco_voice Dokumentation

Inhaltsverzeichnis

1. Vorbereitung.....	2
2. Verwendung.....	3
2.1 Usage / Parameter.....	3
2.2 Modi.....	5
2.2.1 Lern-Modus.....	5
2.2.2 Interface-Modus.....	6
2.2.3 Gesamtstatus.....	8
2.2.4 Nagios-Konfigurations-Modus.....	9
2.3 Optionale Parameter.....	12
2.3.1 Nagios Parameter.....	12
2.3.2 Logging.....	13
2.3.3 Dateipfade.....	13

1. Vorbereitung

Es muss lediglich Python 2.4, Python 2.5 oder Python 2.6 auf dem System installiert sein, um `check_cisco_voice` ausführen zu können. Alle Bibliotheken, die benötigt werden um das Programm auszuführen, sind im Paket selbst mit enthalten. Es muss nichts weiter installiert werden.

Eventuell muss die Datei `check_cisco_voice.py` noch ausführbar (`chmod`) gemacht werden mit folgendem Befehl:

```
chmod u+x check_cisco_voice.py
```

2. Verwendung

2.1 Usage / Parameter

Usage: check_cisco_voice.py -H <host> -c <community> (--in | --out) (-I | -l | -a) [options]

Options:

- version show program's version number and exit
- h, --help show this help message and exit
- l, --learn start voice in learn mode
- I INTERFACE, --interface=INTERFACE
returns the status a single interfaxe
- a, --all returns the whole status of all ports
- i, --in get status of incoming calls
- o, --out get status of outgoing calls
- H HOST, --host=HOST
set hostname or ip
- c COMMUNITY, --community=COMMUNITY
set community
- n, --nagios-config automatic nagios config creator

- C CRITICAL, --critical=CRITICAL
set critical value for nagios
- W WARN, --warn=WARN
set warn value for nagios

- L LOGFILE, --log-file=LOGFILE
set path of log file (default log/HOSTNAME.log)
- log-level=LOGLEVEL
set level (notset|debug|info|warning|error) (default info)
- port-data-file=PORTDATAFILE
destination of port data file (default status/HOSTNAME.portdata)
- port-status-file=PORTSTATUSFILE
destination of port status file (default status/HOSTNAME.portstatus)
- lock-file=LOCKFILE

destination of lock file (default lock/HOSTNAME.lock)

-d, --debug start voice in debug mode (max output)

-t TIMEOUT, --timeout=TIMEOUT
in seconds; after reaching this timeout the procedure
will fail and UNKNOWN will be returned to nagios

-P PORTTTL, --port-ttl=PORTTTL
time to live for volatile status data

-p PORT, --port=PORT
TCP port of agent (default 161)

2.2 Modi

check_cisco_voice kann in 4 verschiedenen Modi aufgerufen werden.

- Lernmodus: wird benötigt um statische Daten zu sammeln und diese persistent abzuspeichern
- Interface-Modus: gibt den Status eines einzelnen Interfaces zurück
- Gesamtstatus: gibt den summierten Status aller Interfaces aus
- Nagios-Konfigurations-Modus: erstellt automatisch durch Fragen an den Benutzer eine Konfigurationsdatei, die in Nagios eingebunden werden kann

2.2.1 Lern-Modus

Arbeitsweise

Im Lernmodus sendet das Programm einen SNMP Aufruf an den Agenten. Dabei wird ein MIB-tree (Baum) nach allen Interfaces abgefragt. Interessant hierbei ist der Interface-Typ. Nur wenn der Interface Typ dem eines ISDN-Interfaces entspricht werden die Daten (ISDN-ID, Port-Nr., ...) weiterverarbeitet.

Im nächsten Schritt wird der ifOperStatus geprüft. Wenn er den Wert 2 hat, so ist das Interface inaktiv und wird nicht weiter betrachtet. Hat der ifOperStatus jedoch einen anderen Wert, so werden die Daten weiter verarbeitet.

Zu den übrig gebliebenen Datensätzen (aktive ISDN-Interfaces) kommen Attribute, wie z.B. die ifDescr hinzu. Nach dem Sammeln der statischen Daten werden sie persistent in eine Datei gespeichert. Den Dateinamen kann man mit folgendem Parameter ändern:

```
--port-data-file=PORTDATAFILE
                        destination of port data file (default status/HOSTNAME.portdata)
```

Der Dateiname entspricht in der Standardkonfiguration dem Hostnamen mit der Endung „.portdata“. Der Inhalt sieht etwa folgendermaßen aus:

```
{'data': {      6: [6, 'E1 0/0/1', 1L, 95L, 2],
                8: [8, 'E1 0/1/1', 1L, 221L, 4],
                10: [10, 'E1 0/2/1', 1L, 347L, 6],
                12: [12, 'E1 0/3/1', 1L, 473L, 8]},
'time': 1236115274.7650001}
```

In der „time“ Variable wird der Zeitpunkt des letzten Aufrufes gespeichert. Im „data“ Block befinden sich die Daten für die einzelnen Interfaces im Stil:

```
interfaceNr : [interfaceNr, ifDescr, ifOperStatus, isdnSignalingIfIndex, isdnPortNr]
```

Aufruf

Der Lernmodus wird mit folgendem Parameter aufgerufen:

```
-l, --learn          start voice in learn mode
```

Es werden dazu folgende Parameter benötigt:

```
-H HOST, --host=HOST
                    set hostname or ip
-c COMMUNITY, --community=COMMUNITY
                    set community
```

Weitere Parameter sind optional, oder beeinflussen den Aufruf in Lernmodus nicht.

Beispiel

Folgender Befehl führt `check_cisco_voice` im Lernmodus aus. Dabei werden die Parameter Hostname (`nrvoice11.aida.de`) und die community (`public`) übergeben:

```
check_cisco_voice.py -H nrvoice11.aida.de -c public -l
```

2.2.2 Interface-Modus

Arbeitsweise

Beim Aufruf des Interface Modus' wird der Status eines einzelnen Interfaces zurückgegeben. Dabei werden die statischen Daten aus der `port-data-Datei`, die im Lernmodus erstellt wurde, geladen. Daher kann der Interface Modus erst aufgerufen werden, wenn vorher der Lernmodus für den Host aufgerufen wurde.

Für jedes Interface, welches sich in der `port-data-Datei` befindet werden folgende Werte in einem einzigen SNMP-Aufruf abgerufen:

```
-isdnLapdOperStatus
-isdnSigStatsInCalls
-isdnSigStatsOutCalls
```


Darüber hinaus müssen noch folgende Parameter übergeben werden:

-H HOST, --host=HOST
set hostname or ip
-c COMMUNITY, --community=COMMUNITY
set community

Außerdem muss noch zwischen eingehenden und ausgehenden Anrufen unterschieden werden. Dies geschieht mit folgenden Parametern:

-i, --in get status of incoming calls
-o, --out get status of outgoing calls

Einer der beiden Parameter muss übergeben werden. Weitere Parameter sind optional oder beeinflussen den Interface Modus nicht.

Beispiel

```
./check_cisco_voice.py -I 8 -H nrvoice11.aida.de -c public -in
```

Dieser Befehl gibt die Anzahl der eingehenden Anrufe pro Minute des Interfaces mit der ID 8 zurück.

2.2.3 Gesamtstatus

Arbeitsweise

Der Gesamtstatus gibt die Summe (der eingehenden/der ausgehenden/aller) Anrufe zurück. Die Arbeitsweise ist äquivalent zum Interface Modus mit dem Unterschied, dass die erhobenen Werte der einzelnen Interfaces summiert zurück gegeben werden.

Aufruf

Der Aufruf geschieht mit folgendem Parameter:

-a, --all returns the whole status of all ports

Es müssen außerdem folgende Parameter übergeben werden:

-H HOST, --host=HOST
 set hostname or ip
-c COMMUNITY, --community=COMMUNITY
 set community

Ohne weitere Parameter wird die Summe aller eingehenden und ausgehenden Anrufe zurück gegeben. Wenn man nur eingehende oder ausgehende Anrufe betrachten möchte, so kann man dies mit folgenden Parametern eingrenzen:

-i, --in get status of incoming calls
-o, --out get status of outgoing calls

Beispiel

```
./check_cisco_voice.py -a -H nrvoice11.aida.de -c public -in
```

Dies gibt die Summer aller eingehenden Anrufe aller Interfaces, auf Minuten gerechnet, zurück.

2.2.4 Nagios-Konfigurations-Modus

Ein Aufruf in diesem Modus erstellt mit Hilfe übergebener Parameter und Fragen an den Benutzer eine Konfigurationsdatei, die in Nagios eingebunden werden kann.

Arbeitsweise

Nahezu alle Parameter, die beim Aufruf in diesem Modus übergeben werden, werden auch in die Konfiguration von Nagios übernommen. Einige Werte können nicht als Parameter übergeben werden und werden daher an den Benutzer gestellt.

Eine mit Platzhaltern versehene Konfiguration wird mit den Werten des Benutzers und den Parametern bestückt und separat abgespeichert.

Aufruf

Bedingung für den Aufruf ist das Vorhandensein der port-data-Datei, welche im Lern-Modus erstellt wird.

Der Aufruf wird mit folgendem Parameter realisiert:

```
-n, --nagios-config    automatic nagios config creator
```

Es müssen außerdem folgende Parameter übergeben werden:

```
-H HOST, --host=HOST
                    set hostname or ip
-c COMMUNITY, --community=COMMUNITY
                    set community
```

Sie werden nach Antworten auf folgende Fragen gefragt:

"Where do you want to store the created configuration file?"

"What's the hostname?"

"Warn-level for single ports (Incoming Calls)?"

"Critical-level for single ports (Incoming Calls)?"

"Warn-level for single ports (Outgoing Calls)?"

"Critical-level for single ports (Outgoing Calls)?"

"Warn-level for all ports (Incoming Calls)?"

"Critical-level for all ports (Incoming Calls)?"

"Warn-level for all ports (Outgoing Calls)?"

"Critical-level for all ports (Outgoing Calls)?"

"Warn-level for all ports (Outgoing + Incoming Calls)?"

"Critical-level for all ports (Outgoing + Incoming Calls)?"

hinter jeder Frage befindet sich in eckigen Klammern [] ein Standardwert, der übernommen wird, wenn der Benutzer nichts eingibt (nur die [Enter]-Taste drückt).

In der Frage nach dem hostname wird der Wert der bestehenden Nagios Konfiguration abgefragt, der den Hostnamen des Agenten wiedergibt. Er muss nicht gleich dem Hostnamen sein, mit dem check_cisco_voice aufgerufen wird.

Die erstellte Nagios Konfigurationsdatei enthält neben der Konfiguration die Versionsnummer und die Erstellungszeit.

Beispiel

```
$> ./check_cisco_voice.py -H nrvoice11.aida.de -n
```

```
Where do you want to store the created configuration file? [/tmp/nrvoice11.aida.de_nagios.cfg]
```

```
Hostname of CISCO SNMP-agent!
```

```
This hostname must be configured in Nagios
```

```
What's the hostname? [nrvoice11.aida.de] nrvoice11
```

```
Warn-level for single ports (Incoming Calls)? [30]
```

```
Critical-level for single ports (Incoming Calls)? [40]
```

```
Warn-level for single ports (Outgoing Calls)? [30] 25
```

```
Critical-level for single ports (Outgoing Calls)? [40] 35
```

```
Warn-level for all ports (Incoming Calls)? [300]
```

```
Critical-level for all ports (Incoming Calls)? [400]
```

```
Warn-level for all ports (Outgoing Calls)? [250]
```

```
Critical-level for all ports (Outgoing Calls)? [350]
```

```
Warn-level for all ports (Outgoing + Incoming Calls)? [550]
```

```
Critical-level for all ports (Outgoing + Incoming Calls)? [750]
```

```
nagios config creation finished
```

2.3 Optionale Parameter

2.3.1 Nagios Parameter

Um manuell Schwellwerte für die Nagios Rückgabe festzulegen können folgende Parameter übergeben werden.

```
-C CRITICAL, --critical=CRITICAL
                        set critical value for nagios
-W WARN, --warn=WARN
                        set warn value for nagios
```

Die Übergebenen Werte entsprechen Anrufe pro Minute. Wenn ein solcher Wert überschritten wurde, wird WARNING, bzw. CRITICAL zurückgegeben.

Beispiel

```
./check_cisco_voice.py -I 8 -H nrvoice11.aida.de -c public -W 20 -C 30
```

Dieser Aufruf gibt den Status WARNING zurück, wenn mehr als 20 eingehende Anrufe pro Minute statt finden. Wenn er Wert 30 überschreitet wird CRITICAL zurückgegeben.

2.3.2 Logging

Um das Logging-Verhalten zu ändern kann man folgende Parameter dazu nutzen:

`-L LOGFILE, --log-file=LOGFILE`

set path of log file (default log/HOSTNAME.log)

`--log-level=LOGLEVEL`

set level (notset|debug|info|warning|error) (default info)

Der Parameter `--log-file` gibt den Pfad zur Datei an, in die Logging-Informationen geschrieben werden sollen. Existiert diese nicht, so wird sie erstellt.

Der Parameter `--log-level` gibt das Log-Level an. Jedes Level schließt das Level darunter mit ein.

Also wenn das Log-Level auf „debug“ gestellt wird, werden auch Warnungen (Level „warning“) ins log geschrieben. Zur Verfügung stehen dabei die Log-Level:

`-notset`: logt alles (selbe Wirkung wie debug)

`-debug`: logt debug Informationen (sollte nicht im Produktionsbetrieb verwendet werden)

`-info`: gibt wichtige Statusdaten zurück (auch positive)

`-warning`: gibt nur Warnungen und Fehler zurück

`-error`: gibt ausschließlich Fehlermeldungen zurück

2.3.3 Dateipfade

Pfad, an dem die statischen Daten, die im Lernmodus erhoben werden, gespeichert werden:

`--port-data-file=PORTDATAFILE`

destination of port data file (default status/HOSTNAME.portdata)

Pfad, an dem status Daten der einzelnen Interfaces gespeichert werden.

`--port-status-file=PORTSTATUSFILE`

destination of port status file (default status/HOSTNAME.portstatus)

Pfad zu einer Datei, die als lock fungiert (geschlossen, wenn Datei vorhanden; offen, wenn Datei nicht vorhanden). Diese Datei wird von `check_cisco_voice` erstellt und gelöscht. Geben Sie kein Pfad zu einer Datei an, die existiert und gegebenenfalls Daten enthält.

`--lock-file=LOCKFILE`

destination of lock file (default `lock/HOSTNAME.lock`)

Pfad zur Logdatei:

`-L LOGFILE, --log-file=LOGFILE`

set path of log file (default `log/HOSTNAME.log`)

2.3.4 Timeout

Mit dem Timeout Parameter kann der Timeout in Sekunden angegeben werden:

`-t TIMEOUT, --timeout=TIMEOUT`

in seconds; after reaching this timeout the procedure will fail and UNKNOWN will be returned to nagios

Wenn eine SNMP Anfrage gesendet wird und eine Zeit lang keine Antwort kommt, wird UNKNOWN/UNREACHABLE an Nagios zurück gegeben. Mit dem Timeout Parameter kann das Verhalten beeinflusst werden. Der Standard Wert liegt bei 60 Sekunden.

2.3.5 Time To Live

`check_cisco_voice` ruft nach dem Ablauf der TTL alle Werte aller Interfaces ab. Nagios kann jedoch immer nur eine Anfrage an ein Interface starten. Daher werden die Werte in einer Datei abgelegt, um zu vermeiden, dass bei jedem Aufruf eine SNMP Anfrage gesendet wird. Wie lange diese Daten gültig sind gibt der Parameter für die TTL an:

`-P PORTTTL, --port-ttl=PORTTTL`

time to live for volatile status data

Der Wert wird in Sekunden angegeben.

2.3.5 TCP-Port

Wenn der Agent nicht auf dem Standard SNMP Port (161) lauscht, so kann dies mit dem Port Parameter geändert werden:

`-p PORT, --port=PORT`

TCP port of agent (default 161)