

# check\_mk

Jens Link

[jenslink@quux.de](mailto:jenslink@quux.de)

FrOSCon 2012

- Freiberuflicher Consultant
- Schwerpunkt: komplexe Netzwerke, Netzwerksecurity, Netzwerkmonitoring, Troubleshooting

**Ich bin käuflich ;-)**

- Open Source Monitoring System
- Ursprünglich: NetSaint
- Der Name setzt sich zusammen aus **Network** und **Hagios** (gr: heilig)
- flexibel, skalierbar, erweiterbar durch Plugin Architektur
- Überwachen läßt sich alles (zu einem bestimmten Grad) was im Netz erreichbar ist
- Entwicklung scheint zu Gunsten der kommerziellen Variante eingeschlafen zu sein :-)

- Nagios Fork
- aktive Entwicklung
- viele Erweiterungen
- gute und aktuelle Doku
- neue GUI
- alte GUI etwas verschönert
- Migration Nagios -> Icinga mittels `ln` und `sed` im Prinzip in wenigen Minuten möglich

- Open Source
- große Community
- Viele fertige Plugins, eigene Plugins lassen sich erstellen
- Viel gute Dokumentation
- Konfiguration über Textdateien

- Für einige: Konfiguration über Textdateien
- Für einige: Keine GUI für die Konfiguration
- Konfiguration kann recht schnell sehr unübersichtlich werden
- SNMP, speziell SNMP Traps
- Monitoring von Netzwerkkomponenten (z.B. “schnell” ein paar hundert Switchports hinzufügen)

- Checks werden über Plugins realisiert
- Man unterscheidet:
  - aktive Checks
  - passive Checks

- übernehmen die eigentliche Überwachung
- können leicht (YMMV) selbst in einer beliebigen Sprache geschrieben werden
- Im einfachsten Fall hat ein Plugin die folgenden Rückgabewerte:
  - 0 - Okay
  - 1 - Warning
  - 2 - Critical
  - 3 - Unknown
- Dazu noch optional: Text und Performance-Daten

Mehr: <http://nagiosplug.sourceforge.net/developer-guidelines.html>



Es gibt bereits zahlreiche fertige Plugins, z.B.:

check_apt	check_disk	check_game	check_imap	check_mailq
check_nt	check_ping	check_simap	check_time	check_bgpstate
check_disk_smb	check_host	check_ircd	check_mrtg	check_ntp
check_pop	check_smtp	check_udp	check_breeze	check_dns
check_hpjd	check_jabber	check_mrtgtraf	check_ntp_peer	check_procs
check_snmp	check_ups	check_by_ssh	check_dummy	check_http
check_ldap	check_mysql	check_ntp_time	check_radius	check_spop
check_users	check_clamd	check_file_age	check_icmp	check_ldaps
check_mysql_query	check_nwstat	check_real	check_ssh	check_wave
check_cluster	check_flexlm	check_ide_smart	check_linux_raid	check_nagios
check_oracle	check_rpc	check_wp	check_dhcp	check_fping
check_ifoperstatus	check_load	check_nntp	check_overcr	check_rta_multi
check_swap	check_dig	check_ftp	check_ifstatus	check_log
check_nntp	check_pgsql	check_sensors	check_tcp	

- nrpe (Nagios Remote Plugin Executor) - Ausführen von Checks auf remote Hosts
- ggf. SSL verschlüsselt
- Einschränkung per IP Adresse (Icinga Variante kann auch IPv6)

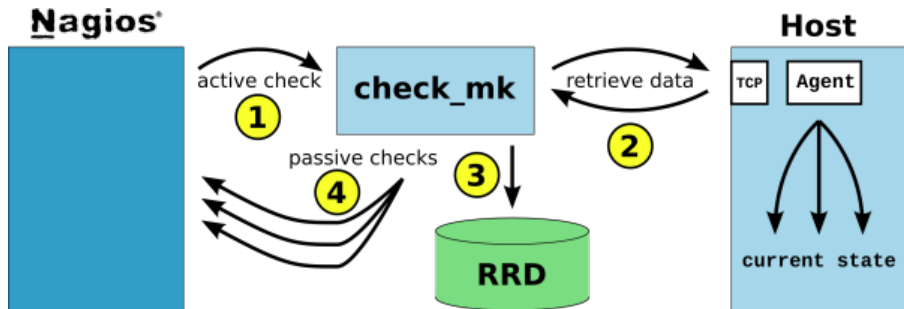
- Daten von passiven Checks über das Netzübertragen
- Verschlüsselung möglich

- Security is Not My Problem
- Simple Network Management Protocol
- Nicht wirklich einfach
- 3 Versionen: 1, 2c, 3
- Nur 3 bietet Verschlüsselung
- OIDs
- MIBs

- OID: Object Identifier
- Beispiel: sys.Uptime = 1.3.6.1.2.1.1.3.
- MIB: Management Information Base

- erleichtert die Erstellen von Nagios/Icinga Konfigurationen
- geschrieben in Python
- eigene Agenten für verschiedenen Betriebssysteme (für \*NIX in shell)
- Monitoring auch per SNMP
- check\_mk erkennt was überwacht werden kann
- Ersatz für NRPE: MRPE

# check\_mk Architektur



Q: [http://mathias-kettner.de/bilder/overview\\_600.trans.png](http://mathias-kettner.de/bilder/overview_600.trans.png)

- 1 Ein aktiver Check pro Host
- 2 check\_mk verbindet sich per TCP (SNMP) mit dem Host und liest alle relevanten Daten aus
- 3 Performance-Daten werden in RRD Files gespeichert (z.B. für pnp4nagios)
- 4 check\_mk extrahiert die Daten, vergleicht diese mit Schwellwerten und übermittelt das Ergebnis als passive check an Nagios/Icinga

- 5 VMs:
  - Monitoring (Icinga + check\_mk)
  - DNS und SMTP
  - Web und FTP
  - Linux Server (nur SNMP)
  - 2. Monitoring Server
- 1 Cisco Router



```
all_hosts = [ 'mon.example.com|demo|ssh|http|https',
              'mail.example.com|demo|ssh|dns|smtp',
              'web.example.com|demo|ssh|dns|smtp|blog',
              'sever.example.com|SNMP|demo|ssh',
              'nagios.example.com|PING|demo|ssh',
              'switch.example.com|SNMP|demo|telnet',
            ]

define_hostgroups = True

host_groups = [
    ( 'DEMO', ['demo'], ALL_HOSTS ),
]
```

# main.mk: Legacy Checks

```
legacy_checks = [  
  ("check_http", "HTTP", True), ["http"], ALL_HOSTS),  
  ("check_https", "HTTPS", True), ["https"], ALL_HOSTS),  
  ("check_ssh", "SSH", False), ["ssh"], ALL_HOSTS),  
  ("check_smtp", "SMTP", False), ["SMTP"], ALL_HOSTS),  
  ("check_imaps", "IMAPS", False), ["IMAP"], ALL_HOSTS),  
  ("check_wp", "Wordpress blog.example.com", False), \  
    ["blog"], ALL_HOSTS),  
]
```

- Es werden OIDs genutzt
- Recht gute Unterstützung für diverse Komponenten
- Flexibel Konfigurationsmöglichkeiten
- Host einfach mit `snmp` taggen

```
snmp_communities = [  
  ( "foobar42", ALL_HOSTS ),  
]
```

```
check_parameters += [  
  ( { "levels" : (85, 90)}, [ "server.example.com" ], \  
    [ "fs_" ] ),  
]  
  
ignored_services = [  
  ( [ "web.example.com" ], [ "Vmalloc address" ] )  
]
```

```
extra_nagios_conf += r"""  
  
define command{  
    command_name      check_wp  
    command_line      $USER1$/check_wp \  
                      --web blog.example.com  
}  
  
"""
```

- Wahlweise aus den Sourcen / als Paket / als .EXE
- \*NIX xinedt (inetd)
- lokale Plugins ausführen: mrpe

```
cat /etc/check_mk/mrpe.cfg  
APT-Status /usr/lib/nagios/plugins/check_apt  
TEMP-BL /usr/local/nagios-plugins/check_eds_tempprobe.py -H
```

- `check_mk -II` - Inventarisierung, erstellen der Nagios Config
- `check_mk -O` - Neustart
- `check_mk -L` - Listet alle möglichen Checks

- pnp4nagios - Performance Werte grafisch darstellen
- livestatus - 1..n Nagios / Icingas unter einer Oberfläche
- icli - GUIs sind für Weicheier! ;-)
- NagiBot - Rede mit deinem Nagios
- check\_v46



eMail	jenslink@quux.de
Jabber	jenslink@guug.de
PGP Fingerprint	D9FF E215 6686 6194 FFC8 A135 19CF A676 DB85 EF91
Blog	<a href="http://blog.quux.de">http://blog.quux.de</a>