check_mk

Jens Link

jenslink@quux.de

sage@guug München 08/2012

Wer bin ich?

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

Ich bin käuflich ;-)

Nagios

- Open Source Monitoring System
- Ursprünglich: NetSaint
- Der Name setzt sich zusammen aus **N**etwork und H**agios** (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 eingschlafen zu sein :-(

Icinga

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

Vorteile gegenüber anderen Systemen

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

Nachteil gegenüber anderen Systemen

- 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)

Nagios Checks

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

Plugins (I)

- 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`

Plugins (II)

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

check_apt
check_nt
check_disk_smb
check_pop
check_hpjd
check_snmp
check_ldap
check_users
check_mysql_query
check_cluster
check_oracle
check_ifoperstatus
check_swap
check_nntps

check_disk
check_ping
check_host
check_smtp
check_jabber
check_ups
check_mysql
check_clamd
check_nwstat
check_flexlm
check_rpc check_ssmtp
check_load
check_dig
check_pgsql

check_game
check_simap
check_ircd
check_udp
check_mrtgtraf
check_by_ssh
check_ntp_time
check_file_age
check_real
check_ide_smart
check_wp
check_nntp
check_ftp
check_sensors

check_imap
check_time
check_mrtg
check_breeze
check_ntp_peer
check_dummy
check_radius
check_icmp
check_ssh
check_linux_raid
check_dhcp
check_overcr
check_ifstatus
check_tcp

check_mailq
check_bgpstate
check_ntp
check_dns
check_procs
check_http
check_spop
check_ldaps
check_wave
check_nagios
check_fping
check_rta_multi
check_log

NRPE

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

NCSA

- Daten von passiven Checks werden das Netz übertragen
- Verschlüsselung möglich

SNMP (I)

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

SNMP (II)

- OID: Object Identifier
- Beispiel: sysUptime = 1.3.6.1.2.1.1.3
- MIB: Management Information Base

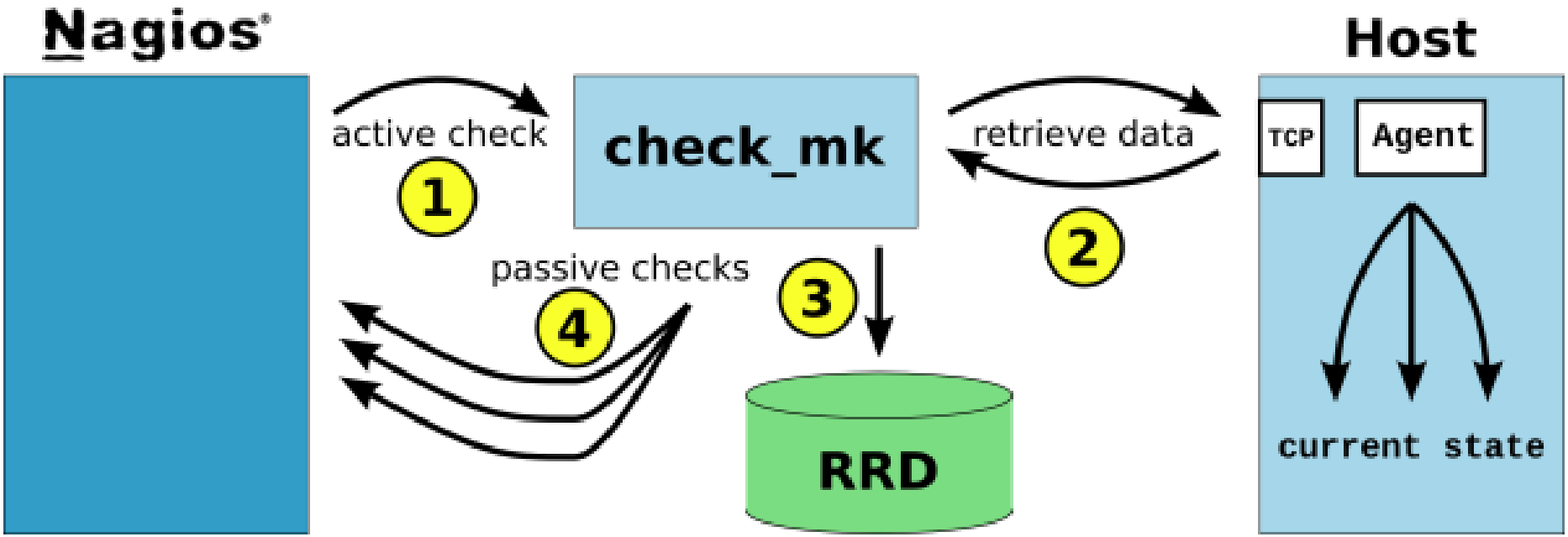
SNMP (III)

- Manager: Stellt SNMP Anfragen, empfängt Traps
- Agent: Beantwortet Anfragen, sendet Traps

check_mk

- erleichtert die Erstellen von Nagios/Icinga Konfigurationen
- geschrieben in Python
- eigene Agenten für verschienden 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`

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

Demo Netz

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

main.mk - Zentrale Konfigdatei

```
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),
]
```

main.mk: SNMP

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

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

main.mk: Schwellwerte und Co.

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

ignored_services = [
   ( [  "web.example.com" ], [ "Vmalloc address" ] )
]
```

main.mk - Klassiche Nagios-Config einbetten

```
extra_nagios_conf += r"""

define command{
        command_name    check_wp
        command_line    $USER1$/check_wp \
              --web blog.example.com
        }

"""
```

Agent

- 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 1-wire -p 2 -w 26 -c 30
```

check_mk Befehle

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

Erweiterungen

- pnp4nagios - Performance Werte grafisch darstellen
- livestatus / multisite - 1..n Nagios / Icingas unter einer Oberfläche
- icli - GUIs sind für Weicheier! ;-)
- NagiBot - Rede mit deinem Nagios
- check_v46
- check_multi - Checks zusammenfassen

check_mk: Multisite

multisite.mk

```
sites = {
  "Foo" : {
      "alias" : "Foo"
  },
  "foo": {
     "alias":          "foo",
     "socket":         "tcp:192.0.2.10:6557",
     "url_prefix":     "http://192.0.2.10/",
   },
}
```

check_v46

```
root@mon:/usr/lib/nagios/plugins# ./check_v46 -H www.guug.de ./check_http
OK: IPv6/www.guug.de OK, IPv4/www.guug.de OK | ipv6_a1_time=0.111082s;;;0.000000
ipv6_a1_size=15281B;;;0 ipv4_a1_time=0.092653s;;;0.000000 ipv4_a1_size=15231B;;;0
Status details:
IPv6/www.guug.de:
HTTP OK: HTTP/1.1 200 OK - 15281 bytes in 0.111 second response time
IPv4/www.guug.de:
HTTP OK: HTTP/1.1 200 OK - 15231 bytes in 0.093 second response time
```

Kontakt

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