

# Mit Nagios das Netzwerk voll im Griff

Philipp Seidel

Chemnitzer StudentenNetz

8.11.2008

# Outline

- 1 Einleitung
- 2 Struktur
- 3 Konfiguration
- 4 Plugin
- 5 Benachrichtigung
- 6 Screenshots
- 7 Ende

# Netzwerkstruktur des Chemnitzer StudentenNetz

- Technik:
  - 6 Server
  - 10 virtuelle Server
  - ca. 80 Switches mit ca. 3700 aktiven Ports
  - zentraler Router
  - 11 W-LAN Accespoints
  - ca. 1800 - 2000 Nutzer (werden nicht geprüft)
- Dienste:
  - SSH
  - Webseite
  - SMTP-Server
  - DNS-Server
  - DHCP-Server
  - Netboot
  - Jabber

# Problem und Lösung

## Problem

- Netzwerkinfrastruktur wächst
- Prüfung von Hosts und Diensten wird schwerer
- frühzeitiges finden von Problemen erschwert

## Lösung

- automatisierte Überwachung

# Nagios

## Vorteile

- sehr leicht konfigurierbar
- leicht erweiterbar
- sehr flexibel
- OpenSource
- verschiedene Benachrichtigungsformen
- Konfiguration in Textdateien

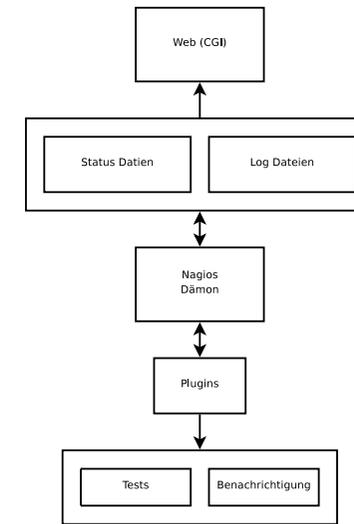
## Nachteile

- gewöhnungsbedürftiges Web-Interface
- Konfiguration in Textdateien

# Was kann überwacht werden?

- Systeme:
  - Linux/Unix System
  - Windows Systeme
  - Routers, Switches, Hubs
  - Drucker
- Dienste:
  - öffentliche Dienste (SSH, HTTP, SMTP, ... )
  - private Dienste (CPU, RAM Nutzung, Festplattenplatz, ... )

# Struktur



# Check Arten

## aktive Checks

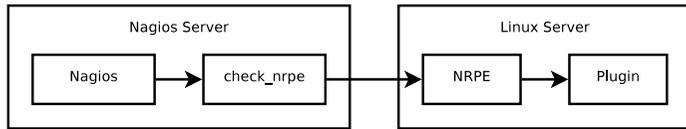
- vom Nagios Daemon ausgelöst
- in regelmässigen Abständen ausgeführt
- Daemon ruft Plugin auf und wertet Rückgabewert aus

## passive Checks

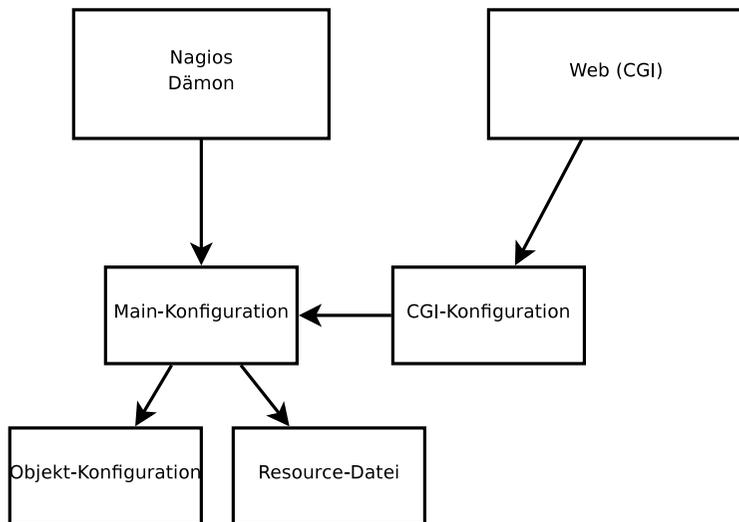
- durch externe Programme durchgeführt
- liefern Status zur weiteren Verarbeitung an Nagios

# private Dienste: Linux/Unix

- zwei grundlegende Methoden
- SSH Verbindung:
  - benutzen von verteilten SSH-Keys
  - Plugin: check\_by\_ssh
  - sehr aufwendig
  - hohe CPU-Auslastung
- NRPE Addon:

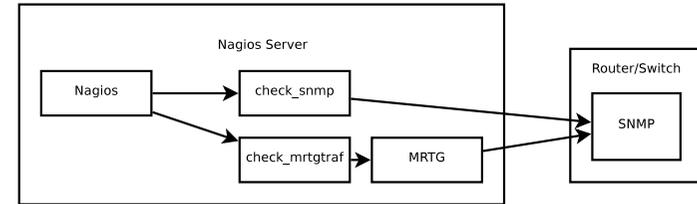


# Struktur der Konfiguration



# private Dienste: Router/Switch

- per SNMP
- per MRTG



# Ordnerstruktur

- lokale Installation in /usr/local/nagios
  - bin - enthält der eigentlichen Nagios Dämon
  - etc - Konfigurationsdateien
  - libexec - Plugins
  - sbin - CGI Skripte
  - share - statische Dateien für die Webseite
  - var - Daten und Log-Dateien
- Paketgebundene Installation

# Konfigurationsdateien

- nagios.cfg - die Hauptkonfigurationsdatei

```

1 ...
2 cfg_file=/usr/local/nagios/etc/objects/commands.cfg
3 ...
4 cfg_dir=/usr/local/nagios/etc/servers
5 ...
    
```

- resource.cfg - Pfadangaben, Passwörter

```

1 ...
2 $USER1$=/usr/local/nagios/libexec
3 ...
4 $USER2$=/usr/local/nagios/libexec/eventhandlers
5 ...
    
```

- cgi.cfg - CGI Konfiguration
- objects/ - weitere Objekte, z.B.: Komandos, Server

# Konfigurationsdateien objects/

- commands.cfg - Kommandos zum Benachrichtigen und um Tests durchzuführen

```

1 ...
2 # 'notify-host-by-email' command definition
3 define command{
4     command_name    notify-host-by-email
5     command_line    /usr/bin/printf "%b" "***** Nagios *****\nNotification Type:
6                     $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress:
7                     $HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /usr/
8                     bin/mail -s "*** $NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$
9                     ***" $CONTACTEMAIL$
10 }
11 ...
12 # 'check-host-alive' command definition
13 define command{
14     command_name    check-host-alive
15     command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w 3000.0,80% -c 5000.0,100% -
16                     p 5
17 }
18 ...
    
```

# Konfigurationsdateien objects/

- contacts.cfg - Personen die Benachrichtigt werden sollen

```

1 ..
2 define contact{
3     contact_name        hans-muster
4     alias                Hans Muster
5     service_notification_period 24x7
6     host_notification_period 24x7
7     service_notification_options w,u,c,r
8     host_notification_options d,u,r
9     service_notification_commands notify-service-by-mail
10    host_notification_commands notify-host-by-mail
11    email                muster@example.com
12    pager                12345678
13 }
14 ...
15 define contactgroup{
16    contactgroup_name    admins
17    alias                Nagios Administrators
18    members              hans-muster
19 }
20 ...
    
```

# Konfigurationsdateien objects/

- localhost.cfg - Checks für Localhost
- printer.cfg - Checks für Drucker
- switch.cfg - Services und Hosts der Switches/Router
- templates.cfg - Templates für Hosts und Services

```

1 ...
2 define host{
3     name                linux-server
4     use                 generic-host
5     check_period        24x7
6     check_interval      5
7     retry_interval      1
8     max_check_attempts 10
9     check_command        check-host-alive
10    notification_period 24x7
11    notification_interval 120
12    notification_options d,u,r
13    contact_groups       admins
14    hostgroups            servers-ping, ssh-servers
15    register              0
16 }
17 ...
    
```

Beispiel Konfiguration  

# Konfigurationsdateien objects/

- timeperiods.cfg

```

1  ...
2  define timeperiod{
3      timeperiod_name 24x7
4      alias            24 Hours A Day, 7 Days A Week
5      sunday           00:00-24:00
6      monday           00:00-24:00
7      tuesday          00:00-24:00
8      wednesday        00:00-24:00
9      thursday         00:00-24:00
10     friday            00:00-24:00
11     saturday          00:00-24:00
12 }
13 ...
    
```

- windows.cfg - Services und Hosts der Windows-Server

Beispiel Konfiguration  

# Konfigurationsdateien servers/

- test-server.cfg (selbst angelegt)

```

1  ...
2  define host {
3      host_name        test-server
4      alias            Test Server
5      address          12.34.56.78
6      parents          test-router
7      use              linux-server
8      hostgroups       +dns-servers
9  }
10 ...
    
```

Beispiel Konfiguration  

# Konfigurationsdateien objects/

- servers.cfg - Services und Hostgroups für Server (selbst angelegt)

```

1  ...
2  define hostgroup {
3      hostgroup_name  servers-ping
4      alias            Pingable servers
5  }
6  ...
7  define service{
8      hostgroup_name    servers-ping
9      service_description PING
10     check_command     check_ping!100.0,20%!500.0,60%
11     use                generic-service
12     process_perf_data  1
13 }
14 ...
    
```

Allgemein  

# Plugin-Aufbau

- Plugin Struktur recht einfach aufgebaut
- Voraussetzung:
  - ausführbar
  - bestimmten Rückgabewert
  - mindestens eine Zeile auf STDOUT ausgeben

Rückgabewert	Service Status	Host Status
0	OK	UP
1	WARNING	UP or DOWN/UNREACHABLE
2	CRITICAL	DOWN/UNREACHABLE
3	UNKNOWN	DOWN/UNREACHABLE

# Erstellen des Plugins

- Plugin muss standardmäßig in Verzeichnis libexec/
- Beispiel: md5sum\_test.sh

```

1  #!/bin/sh
2
3  HOSTNAME=$1
4  FILE=$2
5  MD5SUM_ORIG=$3
6
7  MD5SUM=$(wget -q -O - "http://$HOSTNAME$FILE" | md5sum | awk '{print $1}')
8
9  if [ "$MD5SUM" = "$MD5SUM_ORIG" ]; then
10     echo "MD5SUM:␣$MD5SUM"
11     exit 0
12 else
13     echo "MD5SUM:␣none"
14     exit 2
15 fi
    
```

# Erstellen des Plugins

- Neues Kommando in der Datei commands.cfg anlegen
- Beispiel:

```

1  define command {
2      command_name md5sum_test
3      command_line $USER1$/md5sum_test.sh $HOSTADDRESS$ $ARG1$ $ARG2$
4  }
    
```

- Neuen Service in der Datei server.cfg anlegen
- Beispiel:

```

1  define service {
2      use generic-service
3      host_name test-server
4      service_description MD5SUM Test
5      check_command md5sum_test!/lit-banner-2008.jpg!97121
6      f38355f64fff92d2e719fa22033
7  }
    
```

# vordefinierte Macros

- gekennzeichnet durch \$MACRO\_NAME\$
- werden durch Nagios durch den entsprechenden Wert ersetzt
- Beispiel:

```

1  define service{
2      check_command check_something!5!10
3      ...
4  }
5  define command {
6      command_name check_something
7      command_line ./something -a $ARG1$ -b $ARG2$
8  }
    
```

- Ergebnis:

./something -a 5 -b 10

# eigene Macros

- in Definition mit mit führendem Unterstrich gekennzeichnet
- Zugriff mit:
  - \$\_HOSTvarname\$
  - \$\_SERVICEvarname\$
  - \$\_CONTACTvarname\$
- Beispiel:

```

1  define host{
2      host_name server
3      _MACADDRESS 00:01:02:03:04:05
4      ...
5  }
6  define command {
7      command_name check_something
8      command_line ./something -m $_HOSTMACADDRESS$
9  }
    
```

- Ergebnis:

./something -m 00:01:02:03:04:05

- Funktionsweise fast wie Plugin
- externem Programm wird zusendende Nachricht übergeben
- E-Mail
- SMS
- Anruf
- Pager
- Jabber
- IRC
- Elektroschocker ;-)

## SMS-Versand per SMS Server Tools 3

- zum Versand der SMS einfach Datei in `/var/spool/sms/outgoing/` anlegen
- Format der SMS-Datei:

```
1 To: 1234567
2
3 Test SMS
```

Datei: `/libexec/send_sms.sh`

```
1 #!/bin/bash
2
3 NUMBER=$1
4 MESSAGE=$2
5
6
7 FILE="/var/spool/sms/outgoing/"$(echo "$MESSAGE-$NUMBER"$(date) | md5sum - | awk '{print $1}')."
8   sms"
9
10 echo "To: $NUMBER" > $FILE
11 echo " " >> $FILE
12 echo $MESSAGE >> $FILE
13 chmod a+rw $FILE
```

Datei: `/etc/objects/commands.cfg`

```
1 define command{
2   command_name notify-service-by-sms
3   command_line $USER1$/send_sms.sh $CONTACTPAGER$ '$NOTIFICATIONTYPE$: $HOSTNAME$:
4     $SERVICEDESC$ is $SERVICESTATE$ ($SERVICEOUTPUT$)'
5 }
6
7 define command{
8   command_name notify-host-by-sms
9   command_line $USER1$/send_sms.sh $CONTACTPAGER$ '$NOTIFICATIONTYPE$: $HOSTNAME$ is
10    $HOSTSTATE$ ($HOSTOUTPUT$)'
```

Einleitung 000 Struktur 00000 Konfiguration 000000000 Plugin 00000 Benachrichtigung 00000 Screenshots 000 Ende 00

Versand von SMS

# Personen konfigurieren

Einleitung 000 Struktur 00000 Konfiguration 000000000 Plugin 00000 Benachrichtigung 00000 Screenshots 000 Ende 00

Versand von SMS

# Demo SMS

Datei: /etc/objects/contacts.cfg

```

1 define contact{
2     contact_name      hans-muster
3     alias              Hans Muster
4     service_notification_period 24x7
5     host_notification_period 24x7
6     service_notification_options w,u,c,r
7     host_notification_options d,u,r
8     service_notification_commands notify-service-by-mail, notify-service-by-sms
9     host_notification_commands notify-host-by-mail, notify-host-by-sms
10    email              muster@example.com
11    pager              12345678
12 }

```

- RECOVERY: test-server is UP (PING OK - Packet loss = 0%, RTA = 0.88 ms)
- PROBLEM: test-server is DOWN (CRITICAL - Host Unreachable (12.34.56.78))

Philipp Seidel Chemnitzer StudentenNetz

Mit Nagios das Netzwerk voll im Griff

Philipp Seidel Chemnitzer StudentenNetz

Mit Nagios das Netzwerk voll im Griff

Einleitung 000 Struktur 00000 Konfiguration 000000000 Plugin 00000 Benachrichtigung 00000 Screenshots 000 Ende 00

Bilder aus dem CSN

# Host-Übersicht

Einleitung 000 Struktur 00000 Konfiguration 000000000 Plugin 00000 Benachrichtigung 00000 Screenshots 000 Ende 00

Bilder aus dem CSN

# Host-Ansicht

Host	Plugin	Status	Last Check	Next Check	Current Load	Current Users	Service	Output
ap-63-kl	PING	CRITICAL	11-07-2008 21:54:49	30d 21h 47m 53s 1/3			PING	CRITICAL - Packet loss = 100%
ap-63-kr	PING	OK	11-07-2008 21:57:42	30d 21h 53m 0s 1/3			PING	OK - Packet loss = 0%, RTA = 0.74 ms
ap-64-kl	PING	OK	11-07-2008 21:54:49	30d 21h 47m 46s 1/3			PING	OK - Packet loss = 0%, RTA = 0.62 ms
ap-64-kr	PING	OK	11-07-2008 21:57:49	30d 21h 52m 53s 1/3			PING	OK - Packet loss = 0%, RTA = 0.62 ms
ap-70-kl	PING	OK	11-07-2008 21:54:49	30d 21h 47m 39s 1/3			PING	OK - Packet loss = 0%, RTA = 0.77 ms
ap-72-2	PING	OK	11-07-2008 21:57:59	30d 21h 52m 46s 1/3			PING	OK - Packet loss = 0%, RTA = 2.51 ms
asnar4	PING	OK	11-07-2008 21:54:49	30d 21h 47m 32s 1/3			PING	OK - Packet loss = 0%, RTA = 0.11 ms
	SSH	OK	11-07-2008 21:58:03	31d 6h 32m 39s 1/3			SSH	OK - OpenSSH_4.7p1 Debian-Bunntul1.2 (protocol 2.0)
chartarbox	PING	OK	11-07-2008 21:56:49	30d 21h 47m 25s 1/3			PING	OK - Packet loss = 0%, RTA = 0.80 ms
	SSH	OK	11-07-2008 21:58:11	31d 6h 32m 32s 1/3			SSH	OK - OpenSSH_4.3p2 Debian-9 (protocol 2.0)
cm-jab3	PING	OK	11-07-2008 22:02:02	0d 4h 32m 54s 1/3			PING	OK - Packet loss = 0%, RTA = 0.57 ms
	SSH	OK	11-07-2008 22:02:38	0d 4h 32m 38s 1/3			SSH	OK - OpenSSH_4.7p1 Debian-Bunntul1.2 (protocol 2.0)
cm-server	DNS	OK	11-07-2008 21:56:49	3d 9h 52m 15s 1/3			DNS	OK 0.030 seconds response time: 1.000 returns csn-server.csn.tu-chemnitz.de
	HTTP	OK	11-07-2008 21:59:49	0d 1h 36m 7s 1/3			HTTP	OK - HTTP/1.1 301 Moved Permanently - 0.002 second response time
	DNS	OK	11-07-2008 21:56:49	30d 21h 47m 11s 1/3			PING	OK - Packet loss = 0%, RTA = 1.30 ms
	SMTP	OK	11-07-2008 22:03:49	3d 2h 50m 32s 1/3			SMTP	OK - 0.013 sec. response time
	SSH	OK	11-07-2008 21:56:49	29d 4h 31m 2s 1/3			SSH	OK - OpenSSH_4.3p2 Debian-9etch3 (protocol 1.99)
vermagd	PING	CRITICAL	11-07-2008 21:58:31	30d 21h 52m 11s 1/3			PING	CRITICAL - Host Unreachable (hermodr.csn.tu-chemnitz.de)
	SSH	CRITICAL	11-07-2008 21:56:49	31d 6h 36m 57s 1/3			SSH	No route to host
labber	PING	OK	11-07-2008 21:58:38	30d 21h 52m 5s 1/3			PING	OK - Packet loss = 0%, RTA = 0.27 ms
	SSH	OK	11-07-2008 21:56:49	31d 6h 36m 50s 1/3			SSH	OK - OpenSSH_4.7p1 Debian-Bunntul1.2 (protocol 2.0)
localhost	Current Load	OK	11-07-2008 22:03:11	31d 6h 31m 59s 1/4			OK	load average: 0.00, 0.00, 0.00
	Current Users	OK	11-07-2008 22:03:02	31d 6h 36m 45s 1/4			USERS	OK - 6 users currently logged in
	HTTP	OK	11-07-2008 22:03:18	30d 22h 41m 51s 1/4			HTTP	OK HTTP/1.0 200 OK - 3832 bytes in 0.002 seconds
	PING	OK	11-07-2008 21:56:49	30d 21h 46m 36s 1/3			PING	OK - Packet loss = 0%, RTA = 0.12 ms
	Root Partition	OK	11-07-2008 22:03:09	3d 10h 46m 59s 1/4			DISK	OK - free space: 7.96 MB (96% inode=89%)
	SSH	OK	11-07-2008 21:56:49	31d 6h 36m 39s 1/3			SSH	OK - OpenSSH_4.7p1 Debian-Bunntul1.2 (protocol 2.0)
	Total Processes	OK	11-07-2008 22:03:24	31d 6h 36m 22s 1/4			PROCS	OK 22 processes with STATE = RZSDT
mbr-33	PING	OK	11-07-2008 21:59:42	30d 21h 51m 30s 1/3			PING	OK - Packet loss = 0%, RTA = 1.33 ms

**Service Information**  
 Last Updated: Fri Nov 7 22:07:42 CET 2008  
 Updated every 90 seconds  
 Nagios® 3.0.3 - www.nagios.org  
 Logged in as nagiosadmin

Service: **DNS On Host CSN Server (csn-server)**  
 Member of: No servicegroups.

**Service State Information**  
 Current Status: **OK** (for 1d 9h 55m 1s)  
 Status Information: DNS OK: 0.030 seconds response time. returns  
 csn-server.csn.tu-chemnitz.de.  
 Performance Data: time=0.030121s;:0.000000  
 Current Attempt: 1/3 (HARD state)  
 Last Check Time: 11-07-2008 22:06:49  
 Check Type: ACTIVE  
 Check Latency / Duration: 0.047 / 0.041 seconds  
 Next Scheduled Check: 11-07-2008 22:16:49  
 Last State Change: 11-06-2008 12:12:41  
 Last Notification: N/A (notification 0)  
 Is This Service Flapping? **N/A** (0.00% state change)  
 In Scheduled Downtime? **NO**  
 Last Update: 11-07-2008 22:07:37 ( 0d 0h 0m 5s ago)

**Service Commands**  
 ✖ Disable active checks for this service  
 ⌛ Re-schedule the next check of this service  
 📄 Submit passive check result for this service  
 🛑 Stop accepting passive checks for this service  
 🛑 Stop checking over this service  
 🛑 Disable notifications for this service  
 📄 Send custom service notification  
 📅 Schedule downtime for this service  
 🛑 Disable event handler for this service  
 🛑 Disable flap detection for this service

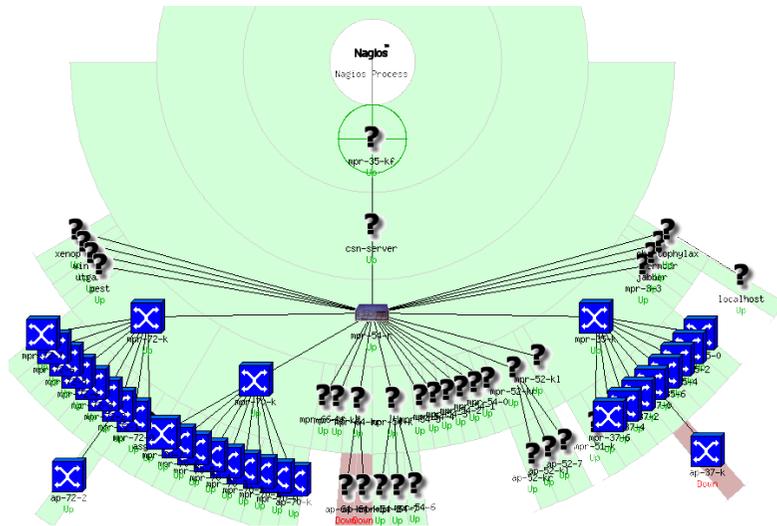
Philipp Seidel Chemnitzer StudentenNetz

Mit Nagios das Netzwerk voll im Griff

Philipp Seidel Chemnitzer StudentenNetz

Mit Nagios das Netzwerk voll im Griff

# Status-Map



# Fragen

# Fragen?

# weitere Informationen

- <http://www.nagios.org/>
- <http://nagiosplug.sourceforge.net/>
- <http://www.nagiosexchange.org/>
- <http://nagioswiki.org/>
- <http://smstools3.kekekasvi.com/>
- <http://www.dinotools.de/>