Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft

DHBW
Duale Hochschule
Baden-Württemberg
Karlsruhe

Entwicklung eines Nagios Plugins zur Überwachung und Auswertung von Funktionen und Fehlern in Content-Managment-Systemen

BACHELORARBEIT

für die Prüfung zum
Bachelor of Engineering

des Studienganges

Informationstechnik

an der Dualen Hochschule Karlsruhe

von

Andreas Paul

Bearbeitungszeitraum:	25.05.2009 – 23.08.2009
Matrikelnummer:	108467
Kurs:	TIT06GR
Praxissemester:	6
Ausbildungsfirma:	Forschungszentrum Karlsruhe GmbH (FZK) Steinbuch Centre for Computing Hermann-von-Helmholtz-Platz 1 76344 Eggenstein-Leopoldshafen
Betreuer:	Dipl.-Ing. (TH) Hans-Peter Hör

Inhalt

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.
Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.
Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

Karlsruhe, den 8. Juni 2009

.....

Ort, Datum (Andreas Paul)

1 Einleitung

Einleitung halt. Kurz was ist Nagios, warum überhaupt überwachen? Was soll überwacht werden -> Stellent/UCM kurz was ist das? Warum gerade das überwachen -> Aktive Benutzung durch User - kritisch

2 Abstract

Zusammenfassung von allem.

Aufgabenstellung, Erwartendes Ergebnis

3 Aufgabenstellung

Oracle UCM werden im FZK eingesetzt, bisher nur rudimentäre Überwachung durch Nagios möglich. Diese Arbeit soll die spezifischen Überwachungselemente erurieren und umsetzten.

4 Grundlagen

In diesem Kapitel werden die grundlegenden Begriffe erläutert, die für das Verständnis der weiterführenden Kapitel notwendig sind.

4.1 ads - ADS Benutzer

4.2 cronjobs

4.3 evtl Oracle DB

4.4 false+-true+-

4.5 Farbraum

4.6 Metadaten

4.7 Nagios

Nagios dient zum Überwachen von Hosts und deren Services in komplexen Infrastrukturen(Host und Services erklären?) und wurde von dem Amerikaner Ethan Galstad seit 1999[1] - damals unter der Vorgängerversion NetSaint - entwickelt und bis heute gepflegt. Galstad gründete aufgrund der vielfältigen(ansturmmäßig) und positiven Resonanz am 9. November 2007 die „Nagios Enterprises LLC“, welche Nagios als kommerzielle Dienstleistung anbietet. Die Software selbst blieb weiterhin unter der freien Lizenz „GNU General Public License version 2“[2] verfügbar. Diese erlaubt Einblick in den Programmcode und Modifizieren der Anwendung nach eigenen Vorstellungen.

Nagios erfreut sich hoher Beliebtheit aufgrund der (bereits vorhandenen [macht

[1] Quelle: `http://www.netsaint.org/changelog.php`

[2] Quelle: `http://www.gnu.org/licenses/old-licenses/gpl-2.0.txt`

kein sinn hohe beliebtheit aufgrund der großen community?]) großen Community, die Tipps, Ratschläge und auch eigene Nagios-Plugins kostenlos anbietet. Außerdem können selbst mit geringen Programmierkenntnissen zusätzliche Skripte zur Überwachung geschrieben werden, wenn ein spezieller Anwendungsfall dies erfordert. Warum wird Nagios engesetzt und nicht was anders -> andere kandidaten finden openview, big brother? -> das buch vom jäger verwenden!

OpenSource halt, recht einfach plugins programmierbar (auf plugin kapitel verweisen)

4.7.1 Aufbau (von Nagios)

Barth schreibt über Nagios:

> „Die große Stärke von Nagios - auch im Vergleich zu anderen Netzwerküberwachungstools - liegt in seinem modularen Aufbau: Der Nagios-Kern enthält keinen einzigen Test, stattdessen verwendet er für Service- und Host-Checks externe Programme, die als *Plugins* bezeichnet werden.“
>
> [Barth08] S. 25

Dieser „Kern“ beinhaltet das komplette Benachrichtigungssystem mit Kontaktadressen und Benachrichtigungsvorgaben (Zeit, Art, zusätzliche Kriterien), die Hosts- und Servicedefinitionen inklusive deren Gruppierungen und schließlich das Webinterface. Siehe hierfür auch Abbildung 1.

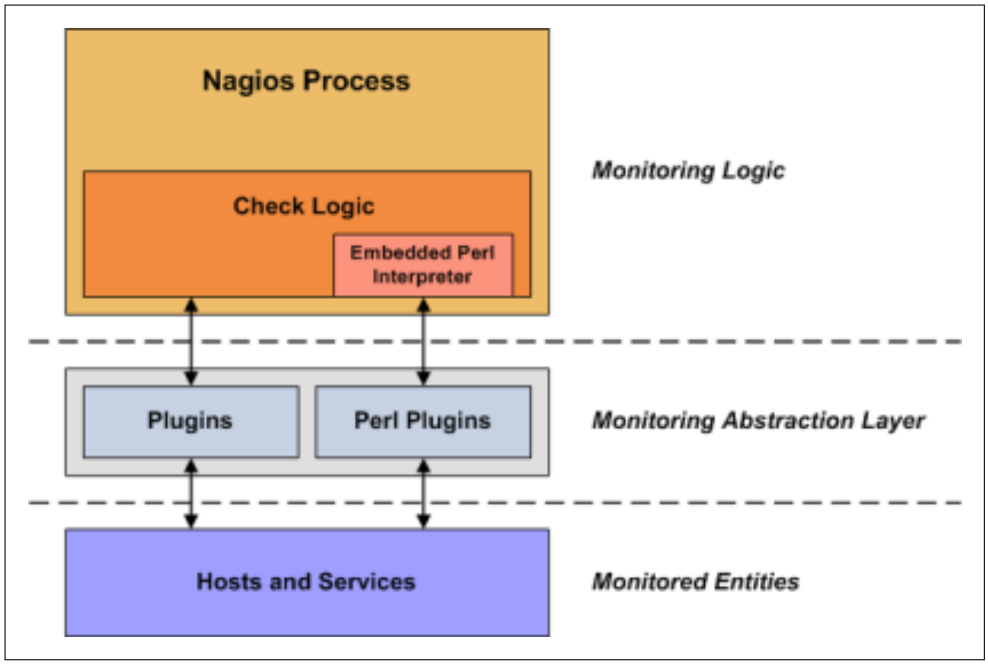

Abbildung 1: Logische Nagios Struktur[3]

[3]Quelle: `http://www.nagioswiki.org/w/images/8/81/Plugins.png`

Die Plugins werden durch die Servicedefinitionen mit den jeweiligen Hosts verbunden und werden durch eine Befehlskonfigurationsdatei mit ggf. veränderten Parametern durch Nagios aufgerufen.
Das bedeutet, dass die gewünschten Plugins explizit aus dem Nagios Repertoire dem zu überwachendem Computer zugeteilt werden müssen. Eine beispielhafte Servicedefinition für den Host *iwrpaul.ka.fzk.de* wird in Codelisting 1.

```
1  # Define a service to check the swap disk space
2  # on the local machine.  Warning if < 20% free,
3  # critical if < 10% free space on swap partition.
4
5  define service{
6   use         generic-service    ;# Name of service template to use
7   host_name   iwrpaul.ka.fzk.de  ;# DNS-Name des Hosts
8   service_description  Swap Disk Space ;# Beschreibung des Services
9   check_command   check_swap!-w 20% -c 10%       ;# Angabe des zu
        verwendenden Plugins mit WARNING (respektiv) CRITICAL
        Schwellwertparameter
10 }
```

Listing 1: Beispielhafte (Definition) eines Servicechecks

Nagios überprüft in einem festlegbaren / veränderbaren Zeitintervall alle vom Benutzer definierten Host- und Servicechecks und verarbeitet / arbeitet / wertet die Daten / Informationen / Ergebnisse der entsprechenden Plugins aus.
Weiterhin beschreibt Barth die Plugins folgendermaßen:

> „Jedes Plugin, das bei Host- und Service-Checks zum Einsatz kommt, ist ein eigenes, selbständiges Programm, das sich auch unabhängig von Nagios benutzen lässt.“
>
> [Barth08] S. 105

Daher lassen sich die Parameter eines Plugins folgendermaßen überprüfen:

```
paul@iwrpaul:/usr/lib/nagios/plugins$ ./check_swap -w 20 -c 10
SWAP OK - 96% free (1826 MB out of 1906 MB) |swap=1826MB;0;0;0;1906
```

Abbildung 2: Beispielhafte manuelle Ausführung eines Servicechecks

Die Ausgabe des Plugins gibt den Zustand des Services an; in diesem Fall wird kein Schwellwert überschritten, daher die Meldung „`SWAP OK`“.
Dabei wird in vier verschiedene Rückgabewerte / Antworten der Plugins unterschieden:

Status	Bezeichnung	Beschreibung
0	OK	Alles in Ordnung
1	WARNING	Die Warnschwelle wurde überschritten, die kritische Schwelle ist aber noch nicht erreicht.
2	CRITICAL	Entweder wurde die kritische Schwelle überschritten oder das Plugin hat den Test nach einem Timeout abgebrochen.
3	UNKNOWN	Innerhalb des Plugins trat ein Fehler auf (zum Beispiel weil flasche Parameter verwendet wurden)

Abbildung 3: Rückgabewerte für Nagios Plugins[4]

Anhand dieser Werte wertet Nagios gezielt den Status des jeweiligen Objektes (Host oder Service) aus. Weiterhin gibt es weiche (Soft States) und harte Zustände (Hard States):

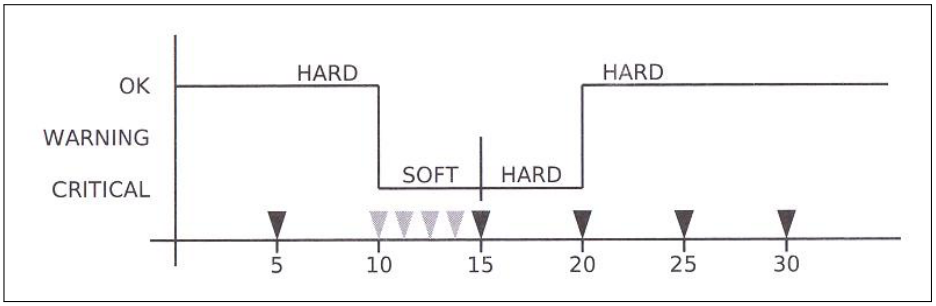

Abbildung 4: Beispiel für den zeitlichen Verlauf durch vers. Zustände[5]

[4]Quelle: [Barth08] S. 105f

Ausgehend von einem OK Zustand wird in diesem Beispiel jede fünf Minuten periodisch überprüft, ob sich der Status des übwachten Objektes verändert hat. Nach zehn Minuten wird ein Umschwenken / Änderung des Zustandes durch das jeweilige Plugin gemeldet

> „; der Zustand wechselt nach CRITICAL, zunächst allerdings als Soft State. Zu diesem Zeitpunkt löst Nagios noch keine Benachrichtigung aus“

, da es sich um eine Falschmeldung, auch False Positive genannt, handeln kann, aufgrund einer kurzweiligen / kurzfristigen (besseres Wort? peak mäßig) hohen Auslastung des Netzwerkes oder um ein kurzzeitiges Problem, welches sich von alleine wieder normalisiert. (Bspw. Prozessorauslastung)
Nun wird der im Soft State befindliche Service bzw. Host mit einer höheren Frequenz überprüft, damit ein False Positive ausgeschlossen werden kann. Sollten diese Überprüfungen den vorherigen Zustand bestätigen, verfestigt sich der aktuelle Zustand, man spricht nun von einem Hard State / wechselt der Zustand in den Hard State. Erst in diesem Moment werden die entsprechenden Kontaktpersonen über den in diesem Beispiel kritischen Zustand benachrichtigt. Sollte sich der Zustand wieder in den Normalzustand begeben und dieser Zustandsübergang wird von dem (von Nagios ausgeführten) Plugin festgestellt, wird dies an den Nagios Server gemeldet.
Ein Übergang zu dem OK Status wird sofort als Hard State festgelegt / festgehalten / festgesetzt / und führt dadurch zur sofortigen Benachrichtigung durch Nagios.

- Betroffene OSI Schichten auflisten und erklären
- Wie werden die Info von Nagios gesammelt und wie gespeichert -> FlapDetection

[5] Quelle: [Barth08] S. 95

- **Performacedaten???**
- FLapping [6]

4.7.2 Service Checks und deren / ihre Realisierung / Ausführung / (Überprüfungs)Methoden

Dienste, die im Netzwerk zur Verfügung stehen (Netzwerkdienste), wie ein Web- oder FTP-Server, lassen sich einfach / simpel direkt über das Netz auf ihren Zustand (hin) überprüfen / testen. Hierfür muss dem entsprechende Plugin lediglich die Netzwerkadresse mitgeteilt werden, siehe Abbildung 5 als beispielhafte Überprüfung eines Webservers.

```
liwrpaul:/usr/lib/nagios/plugins$ ./check_http -H scc-bm-01.scc.kit.edu
OK HTTP/1.1 200 OK - 5194 bytes in 0.008 seconds |time=0.008195s;;;0.000000 size=5194B;;;0
```

Abbildung 5: Beispielhafte manuelle Ausführung eines netzwerkbasierenden Servicechecks / HTTP Server Check

(Bitte beachten, dass das Plugin immernoch auf dem Nagios Server ausgeführt wird / sich immernoch auf dem Nagios Server befindet)
Dienste, die sich nicht standardmäßig / ohne weiteres / ohne weitere Anpassung(en) über das Netzwerk überprüfen lassen, wie die Kapazität einer Festplatte auf einem entfernten Server(, das (Laufen) eines Prozesses) oder die Durchsuchung einer Logdatei nach bestimmten (Stop)wörtern.
Nagios bietet verschiedene Möglichkeiten an solche Dienste / Services zu überprüfen:
Hier die Zahlen als Wort oder Ziffer stehen lassen?

Variante / Methode / Client 1 Der zuvor, in Abbildung 5, gezeigte / abgebildete Test eines netzwerkbasierenden Dienstes wird im obigen Bild

[6] `http://nagios.sourceforge.net/docs/2_0/flapping.html`

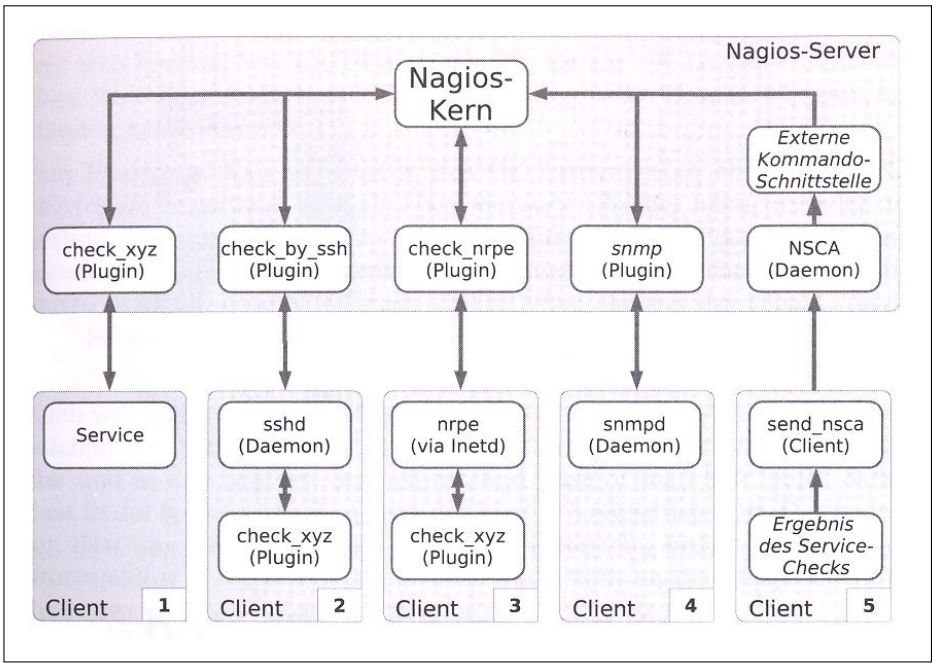

Abbildung 6: Verschiedene Überwachungsmöglichkeiten von Nagios [7]

mit dem Client-Rechner (mit der Nummer) 1 realisiert. Die Überprüfung von nicht netzwerkbasierenden Diensten soll mit den restlichen Client-Rechnern exemplarisch aufgezeigt werden.

Variante / Methode / Client 2 Falls es sich beim Client um ein Unix-derivat handelt, ist der entfernte Zugriff auf diesen Client per SSH[8]-Dienst möglich. Dazu muss auf dem Client ein SSH-Benutzerkonto angelegt sein, mit dem sich Nagios anmelden kann und die öffentlichen Schlüssel (zwischen Nagios Server und Client) ausgetauscht werden, damit keine passwortabhängige Benutzerauthentifizierung (Eingabe von PW) notwendig ist. Danach können lokale Ressourcen, wie Festplattenkapazität oder Logdateien mit dem entsprechenden Plugin direkt auf dem entfernten Rechner überwacht werden. Damit der Client diese Plugins verwenden kann, müssen sich die gewünschten

[8]Durch eine Secure Shell (SSH) kann man sich eine verschlüsselte Netzwerkverbindung zum entfernten Rechner aufbauen.

Plugins (auch) auf dem Client (lokal) befinden. Eine beispielhafte Verwendung mit dem dafür gedachten Nagios Plugin „`check_by_ssh`“ (von dieser Überwachungsmethode) wird in Abbildung 7 gezeigt.

```
paul@iwrpaul:/usr/lib/nagios/plugins$ ./check_by_ssh -H ppt.ka.fzk.de -C "/bin/check_swap -w 20 -c 10"
SWAP OK - 100% free (384 MB out of 384 MB) |swap=384MB;0;0;0;384
```

Abbildung 7: Beispielhafte manuelle Ausführung eines Servicechecks über SSH

(Hier beachten, dass kein Passwort abgefragt wird, daher zuvor Schlüsselaustauschen)

Variante / Methode / Client 3 Eine alternative Möglichkeit solche Dienste auf entfernten Rechnern zu überwachen, ist durch den sogenannten Nagios Remote Plugin Executor (kurz NRPE). Hier muss auf dem Client (extra) ein „Agent“ installiert werden, welcher einen (TCP)-Port öffnet und auf diesem auf Anweisungen durch den Nagios Server horcht.
Der Nagios Server kann diese Anforderungen über das (dafür gedachte) Plugin „`check_nrpe`“ an den Client verschicken. Ein Aufruf dieses Plugins ist dem des „`check_by_ssh`“ Plugins, siehe dazu Abbildung 7, sehr ähnlich.
Der Nachteil dieser Variante ist ein zusätzlich geöffneter Port und der höhere / erhöhte Aufwand beim Installieren des Agenten im Gegensatz zum (vermutlich /meistens) bereits laufendem SSH-Dienst. Zusätzlich gibt es nur die Möglichkeit die Anfragen auf diesem Port auf bestimmte IPs zu beschränken, jedoch nicht den Zugriff durch ein Passwort zu sichern. Dafür beschränkt sich der NRPE (lediglich) auf die auf dem entfernten Client liegenden Nagios Plugins und kann nicht System- bzw. Benutzerkommandos aufrufen, wie bspw. das „`rm`“ Kommando zum Löschen von Dateien, welches durch den Einsatz von „`check_by_ssh`“ standardmäßig möglich wäre. Daher könnte die SSH Variante (wenn unbehandelt PATH, user einschränken) fatale Folgen

nach sich ziehen, wenn es ein Angreifer schafft die Kontrolle über den Nagios Server zu erlangen. Beide Verfahren (hingegen) unterstützen die Verschlüsselung des Datenaustausches untereinander.

Variante / Methode / Client 4 Diese Variante wird nur grob angerissen / kurz / verkürzt behandelt, da sich diese Arbeit hauptsächlich mit der Überwachung von Servern beschäftigt und nicht von Netzwerkkomponenten wie Switches oder Router, die nur per SNMP überwacht werden können, wenn mehr Informationen als eine schlichte Erreichbarkeit überprüft / gesammelt werden soll.
Barth schreibt über diese Variante / Überwachungsmethode:

> „Mit dem Simple Network Management Protocol SNMP lassen sich ebenfalls lokale Ressourcen übers Netz abfragen [...]. Ist auf dem Zielhost ein SNMP-Daemon installiert [...] kann Nagios ihn nutzen, um lokale Ressourcen wie Prozesse, Festplatten oder Interface-Auslastung abzufragen."
>
> [Barth08] S. 101

Durch SNMP kann auf die strukturierte Datenhaltung der MIB[9] in den entfernten Netzwerkknoten zugegriffen werden.

TODO: SNMP MIB?

Man unterscheidet zwischen aktiven und passiven Checks.

[9]Die Management Information Base (MIB) dient als SNMP-Informationstruktur und besteht aus einem hierarchischen, aus Zahlen aufgebauten Namensraum. Ähnliche Struktur wie andere hierarchische Verzeichnisdiensten wie DNS oder LDAP. Quelle: [Barth08] S.233

passive asynchron! Bei passiven Tests führt der zu überwachende Computer das statuserzeugende Ergebnis selbst aus und sendet es über ein Plugin zum Nagios Server. Hierfür muss das Testprogramm bzw Script und das entsprechende Plugin „`send_ncsa`“, welches zum Versenden der Informationen zuständig ist, auf dem Host vorhanden sein. Auf der anderen Seite muss der „`NSCA`“ (Nagios Service Check Acceptor) als Dämon gestartet sein, damit die übermittelten Ergebnisse entgegengenommen werden können.

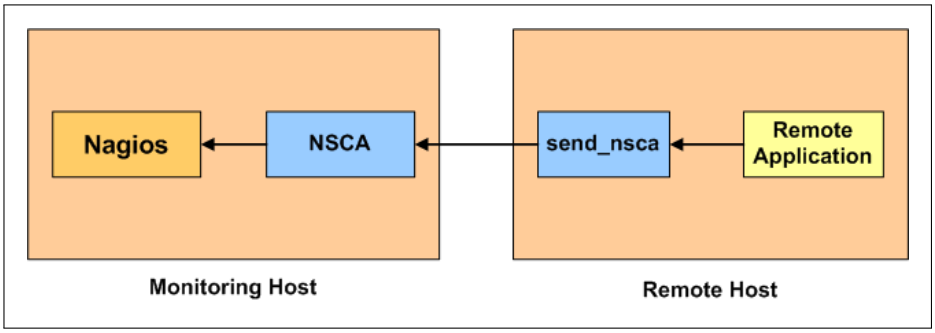

Abbildung 8: Passive Checks mit NSCA

10

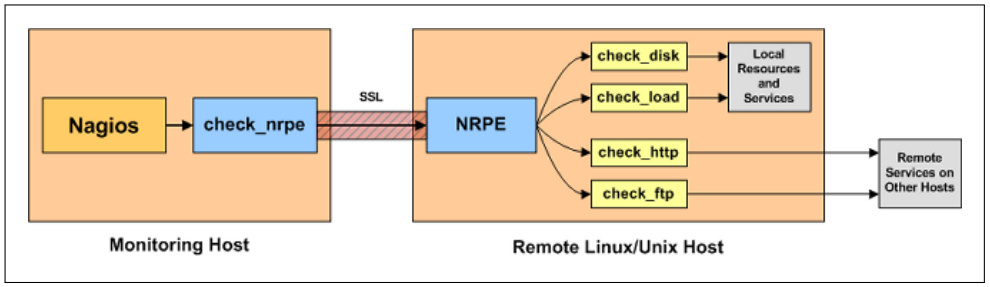

Abbildung 9: Aktive Checks mit NRPE

11

- Kurz agenten, zeigen auf f. Kapitel -> SNMP erklären (MIB, OID) Sicherheitsrisiko

4.7.3 Konfigurationsdateien?

- Wie/wo werden Hosts eingetragen -> hostgroups

- Service Definitionen -> mit Host verbinden
- Wer wird wann wegen was wie benachrichtigt -> contacts

4.7.4 Plugins

Gedacht für Linux umgebung

Verschiedene Möglichkeiten Checks zu realisieren unter Unix Systemen:

	SSH	NRPE	SNMP	SNMP traps	NSCA
Connection initiation	Srv -> Clnt	Srv -> Clnt	Srv -> Clnt	Clnt -> Srv	Clnt -> Srv
Security	Encryption TCP wrappers Key pairs	Encryption Access List TCP wrappers	Access List (v2) Password (v3)	Access List (v2) Password (v3) TCP wrappers	Encryption Access List TCP wrappers
Configuration	On server	On client	On client	On client and On server	On client
Difficulty	Easy	Moderate	Hard	Hard	Moderate

Abbildung 10: Übersicht der verschiedenen Unix Agenten[12]

Leicht programmierbar -> perl Extra Plugins für Windows

4.7.5 (Windows) Agenten oder allgemein Einholen von Infos

Warum nicht einfach alles über SNMP? -> ODI muss man erst beantragen, hoher Aufwand und dann doch nicht so universell/alles abdeckend wie aktive checks, man kann keine logfiles durchuchen -> könnte es aber als standalone prog auf dem client laufen lassen und dieser sendet dann passive checks

Sagen das auf alten NSClient verzichtet wird und OpMon Agent nicht behandelt

1. Bilder ausm Nagios Buch Seite 472ff!

[12]Quelle: `http://www.kilala.nl/Sysadmin/index.php?id=708`

2. NSClient++
3. NC_Net
4. NRPE_NT

Zusammenfassung?

	NSClient	NRPEnt	NSClient++	SNMP	SNMP traps	NC_net **
Connection initiation	Srv -> Clnt	Srv -> Clnt	Srv -> Clnt	Srv -> Clnt	Clnt -> Srv	Clnt -> Srv Srv -> Clnt
Security	Password	Password Encryption	Password Encryption * ACL	Access List Password	Access List Password	Encryption ACL
Configuration	On client	On client	On client	On client	On client and On server	On client
Difficulty	Moderate	Moderate	Moderate	Hard	Hard	Moderate
Resource usage ***	unknown	unknown	9MB RAM	unknown	unknown	30MB RAM

Abbildung 11: Übersicht der verschiedenen Windows Agenten

13

Welche wird jetzt eingesetzt und warum?

Erwähne sichheitsstechnisch Parameter erlauben oder nicht erlauben

Dabei sagen, dass wenn nicht erlaubt sind keine zentrale Konfiguration der Checks auf dem Nagios server möglich ist -> abwägen

4.8 Stellent/Oracle UCM

UCM CMS erklären/erläutern

Oracle DB als Grundlage

FZK Bilddatenbank erwähnen Inbound refinery

Bild **??** S 17

Sessionanzahl
Für was wird sie im FZK benutzt, auch Windows nennen
Einchecken -> Konvertieren -> Indexieren

4.9 Überwachungselemente

Die Überwachung einer Dienstes über ein Netzwerk verteilt sich auf verschiedenen Ebenen mit unterschiedlichen Gewichtungen. Zum Beispiel stellt das simple Senden eines Pings an den entsprechenden Server die niedrigste und primitivste Stufe dar, da hier lediglich die Netzwerkschnittstelle des Servers auf ihre Funktionalität und dabei der Status der Netzwerkstrecke getestet wird.(Rechner an und Netzwerk ok) Ob die Anwendung überhaupt auf dem Server läuft und wenn, in welchem Zustand sie sich befindet (betriebsfähig, reagiert nicht mehr usw.), muss auf eine andere Weise herausgefunden werden.
Dabei lässt sich aus den verschiedenen Überwachungselemente folgende drei Kategorien bilden/ableiten:

4.9.1 Statusabfragen

Diese Kategorie besteht aus einfacheren Überprüfungen, die jeweils den Status des Überwachungselementes überwachen. Dabei können weitere Untergruppen gebildet werden.
(Irgendwie Hierarchie verdeutlichen: zuerst ping als Grundlage, dann prozesse und services dann funktionschecks, bloss wie? Visio Bild?)

System

- **Ping** Überprüft, ob der Rechner vom Nagios-Server über das Netzwerk erreichbar ist.
- **Prozessorauslastung** Überwacht die Auslastung des Prozessors und schlägt bei ungewöhnlich hohen Werten Alarm.
- **Festplattenspeicherausnutzung** Überwacht die Speicherplatzauslastungen der verschiedenen Festplattenpartitionen, damit immer genügend Speicherplatz für Anwendungen und Betriebssystem verfügbar ist.
- Temperatur ??? Die ganzen Standardüberwachungen -> Thüllmann
- **Sessionanzahl** Anzahl der am CMS angemeldeten Benutzer, da aus Performanzgründen eine Obergrenze mit einer maximalen Anzahl festgelegt ist.

Prozesse

- **IdcServerNT.exe** Der Windowsprozess des Stellent-Servers
- **IdcAdminNT.exe** Der Windowsprozess für die Administration (Webinterface?) des Stellent-Servers
- **w3wp.exe** Der Windowsprozess des Microsoft „Internet Information Services“

Services

- **IdcContentService???** Den Zustand des „Content-Dienstes“-„`sccdms01`“ überprüfen.
- **IdcAdminService???** Den Zustand des „Administrations-Dienstes“-„`sccdms01_admin`“ überprüfen.

- **Zeitsynchronisationsdienst:** Überprüfen, ob der „`W32TIME`“-Dienst, der für den Zeitabgleich mit einem Zeitserver zuständig ist, läuft und die Abweichung zwischen Client und Zeitserver festhalten.
- **Antivirusdienst**: Den Zustand den Dienstes überprüfen, der für die ständig Updates des Virusscanners Symantec AntiVirus notwendig ist.

4.9.2 Überwachung der Funktionalität

Durch die vorherigen Tests kann herausgefunden werden, ob eine Anwendung oder ein Dienst auf dem Server gestarten wurde. Die Funktionalität kann durch solche Überprüfungen jedoch **nicht** sichergestellt werden. Da beispielsweise der Prozess bzw. Dienst des Webservers gestartet ist, jedoch keine Webseite aufgerufen werden kann. Daher muss eine weitere Art von Überprüfungen/Checks die Anwendungen auf ihre Funktionalität (hin) überprüfen.

- **Webserver** Aufruf einer Webseite auf dem Server. Wenn auf diese Anfrage eine gültige Antwort in Form einer Statuscode-Meldung erfolgt, kann der reale/wirkliche Zustand des Webservers festgestellt werden.
- **Webinterface des Oracle UCM** Zusätzlich wird mit dieser Abfrage die Integration des Conten-Management-Systems in den Webserver überwacht, da hier nicht nur der Webserver, sondern eine UCM spezifische Webseite abgefragt wird.
- **Benutzeranmeldung am Oracle UCM** Hier wird getesten, ob sich ein Benutzer erfolgreich am System anmelden kann. Dies wird mit Anmeldungsdaten eines lokalen Benutzers und eines Active Directory-Benutzers durchgeführt um gleichzeitig/zusätzlich die Verbindung zum ADS-Server zu testen.
- **Oracle Datenbank** Wenn keine Verbindung zur Oracle Datenbank möglich ist, können keine neuen Informationen gespeichert werden. (zi-

tat riester: läuft das system noch pseudomäßig, wirft aber jede Menge Fehler)

- **Status von Cronjobs** In periodischen Zeitabständen werden Programme aufgerufen, deren Aufruf und Endstatus/Endergebniss überwacht werden muss. Damit nicht das vorherige Ergebnis zu einem False Negative führt, müssen hier zusätzliche Zeitinformationen/zeitliche Parameter beachtet/bedacht werden.

- **Einchecken von Dokumenten** Damit die eigentliche Aufgabe des Dokumentenverwaltungssystem überwacht werden kann, werden verschiedene Datenformate testweise eingecheckt. Dabei wird die Antwort der Anwendung auf das Hinzufügen der Dateien analysiert.

- **Konvertierung** Da das hinzugefügte Dokument nicht nur einfach auf dem Server gespeichert wird, sondern dabei auch in ein anderes Format umgewandelt wird, muss diese Konvertierung zusätzlich überwacht werden. Wird beispielsweise ein Bild eingecheckt, wird dieses mehrfach in verschiedenen Auflösung oder als anderes Bilddateiformat gespeichert. Ob diese Transformation erfolgreich ablief, kann anhand dieser neuen Dateien festgestellt werden.

- **Indizierung** Bei dem Einchecken sollen auch gleichzeitig zusätzliche Informationen über das Dokument festgehalten werden. Diese Informationen können beispielsweise der Name des Authors, das Erstellungsdatum der Datei oder - bei Bildern - der verwendete Farbraum sein. Bei der Suche nach einem Dokument können diese Informationen als zusätzliche Suchkriterien verwendet werden. Daher müss überprüft werden, ob diese Daten richtig ausgelesen werden, der Datenbank hinzugefügt und vom Anwender abgefragt werden können. Dabei werden auch zuvor festgelegte/ausgewählte Testdateien verwendet.

4.9.3 Auswerten von Logdateien

Die zwei bisherigen Kategorien beinhalten simple Zustandsüberprüfungen oder aktive Funktionaltests. In dieser Kategorie werden zusätzlich verschiedene Logs auf spezielle Warnungs- und Fehlermeldungen anhand entsprechender eindeutigen Signal/Stopwörter untersucht. Dies ist notwendig um Fehlverhalten der Anwendung zu erkennen, das nicht mit den vorherigen Überwachungselementen entdeckt wurde. Desweiteren können durch die Analyse der Logdateien etwaige Alarmmeldungen der bisherigen Tests bestätigt, begründet oder aufgehoben werden. Somit bietet das Auswerten der Logdateien zusätzliche Sicherheit False Positive- oder False Negative-Meldungen auszuschliessen.
Die Oracle UCM Anwendung erstellt drei verschiedene Arten von Logdateien:[14]

- **Content Server Log**
- **Inbound Refinery Log**
- **Archiver Log**

Um alle Logs ohne Probleme im Internetbrowser anzuzeigen, liegen alle Logdateien im HTML-Format vor. Alle drei Arten von Logs bestehen jeweils aus 30 verschiedenen Dateien, die sich täglich abwechseln. Dadruch wird für jeden Tag im Monat eine separate Datei angelegt, um bei vielen Warnungs- und Fehlermeldungen durch die chronologische Anordnung/Hierarchie den Überblick zu behalten. So werden die Logdateien zwangsweise nach 30 Tagen nacheinander überschrieben.
Diese Rotation der Logdateien muss bei der Durchsuchung nach Signal/Stopwörter beachtet werden, damit stets die aktuelle Logdatei überwacht wird

[14]Quelle: [UCMlog09]

und keine veralteten Informationen für False Positives-Meldungen durch Nagios sorgen.

Übersichttabelle? wie mit beschreibung verbinden?

5 Abbildungsverzeichnis

6 Literaturverzeichnis

[DMS08] Götzer, Klaus; Schmale, Ralf; u.a. (2008) „Dokumenten-Management - Informationen im Internehmen effizient nutzen“ 4. Auflage,
dpunkt.verlag GmbH Heidelberg, ISBN13: 978-3-89864-529-4,
Stand: ????, Einsichtnahme: 05.06.2009

[Barth08] Wolfgang Barth (2008) „Nagios - System- und Netzwerk-Monitoring“ 2. Auflage,
ISBN13: 978-3-937514-46-8,
Stand: ????, Einsichtnahme: 14.05.2009

[Huff06] Brian Huff (2006) „The Definitive Guide to Stellent Content Server Development“,
ISBN13: 978-1-59059-684-5,
Stand: ????, Einsichtnahme: 15.05.2009

[UCMlog09] Unbekannter Author „vramanat“ (2009) „Universal Content Management 10gR3 - Content Server Log File Information“,
Quelle: `http://www.oracle.com/technology/products/content-management/cdbs/loginfo.pdf`
Stand: 20.01.2009, Einsichtnahme: 05.06.2009