swest

SVISOR

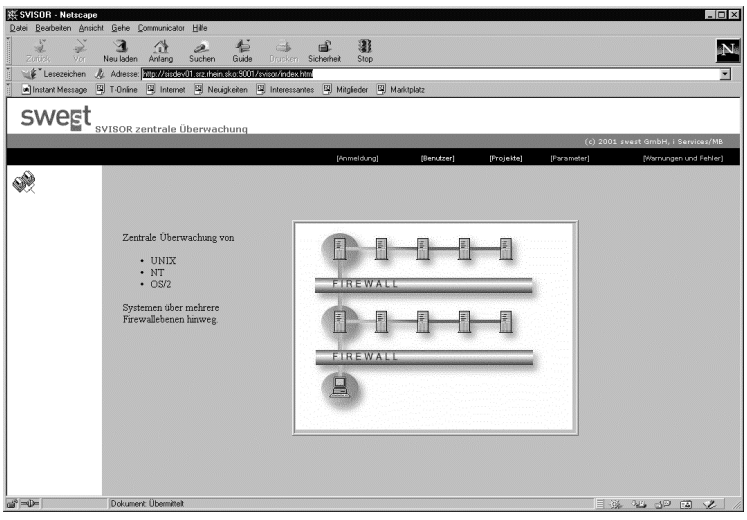
zentrale Überwachung

Vortrag für die GUUG von Malte Beißel

swest GmbH
Malte Beißel
Edmun -Rumpler-Str. 6c
51149 Köln
Tel.: 02203/5039-3241
eMail: malte.beissel@swest.de

Zentrale Überwachung

swest



swest

Zentrale Überwachung

- Anforderungen an ein Überwachungssystem
- Sicherheitsbetrachtungen
- Lösung
  - Prinzipielle Funktionsweise
  - Die Datenbank
    - auf dem Master
    - auf dem Collector
  - Datenaustausch
  - Alarmierung
- Ausblick

swest

Anforderungen an ein Überwachungssystem

- Einfache Handhabung
- Sicher
- Borwserbasierend
- Plattformunabhängig
- Übersichtliche Darstellung der überwachten Maschinen, auch wenn es viele sind
- Alarmierung per eMail und SMS
- Überwachung auch per SNMP ermöglichen
- Fehlerhistory (wann stand welcher Dienst nicht zur Verfügung?)
- Grafische Auswertung von Parametern (z.B. für Trendanalysen)
- Zentrale Konfiguration für alle überwachten Systeme in allen Netzsegmenten
- Mandantenfähig
- Erweiterbar
- Preisgünstig

swest

Anforderungen an ein Überwachungssystem

Überwachung von:

- Uptime
- CPU-Last
- Speicherauslastung
- IO-Wait
- Anzahl der Prozesse
- Plattenplatz
- Dateigröße
- Existenz eines bestimmten Prozesses
- Funktion eines bestimmten Dienstes (z.B. Oracle-DB oder Webserver)
- Datenvolumen auf den Interfaces (Net-Traffic)
- .....

swest

Sicherheitsbetrachtungen: Netz

Die Sicherheitsanforderungen eines Unternehmens müssen berücksichtigt werden!

Einige Hersteller haben sehr lange gebraucht, um das zu verstehen .....

In einem sicheren Netzwerk sind folgende Randbedingungen zu erfüllen:

- Kein transparentes Durchrouten von Datenpaketen durch DMZs
- Keine bidirektionaler Datenverkehr zwischen den DMZs
- Daten dürfen nur in die Richtung des „Feindes" gesendet werden

Weitere Einschränkungen:

- Kein SNMP-Verkehr zwischen den DMZs

swest

Sicherheitsbetrachtungen: Server

Die Sicherheitsanforderungen eines Unternehmens müssen berücksichtigt werden!

Einschränkungen:

- Die Überwachungsdienste dürfen keine Funktionen bereitstellen, die den Betrieb des Servers gefährden können.

- kein Dienst, der für die Überwachung erforderlich ist, darf „Admin-Rechte" erhalten.

swest

Gibt es so etwas nicht schon?

Ja .... aber

- Entweder kostenlos, dann aber nicht so umfangreich und flexibel

- Kommerziell
  - teuer
  - viele unnötige Funktionen
  - hoher Trainingsbedarf

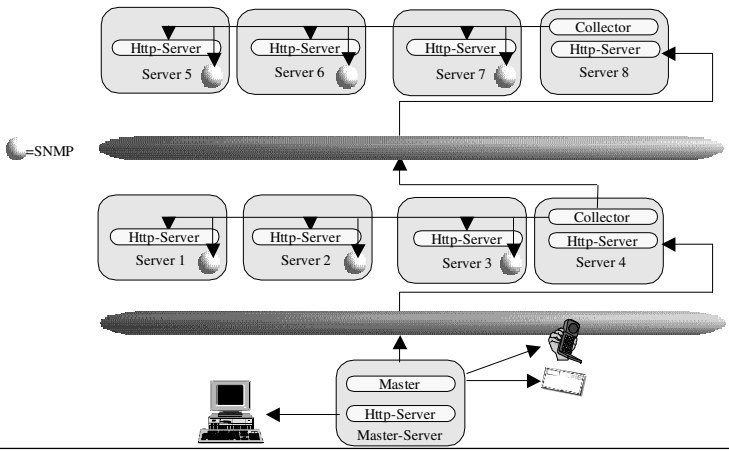
sweet

Die Lösung

..... Eigenentwicklungen .....

sweet

Lösung



Eingesetzte Software

- Auf den Collectoren wird installiert:
  - mysql oder andere DB
  - apache
  - php
  - cUrl
  - JRE
  - JDBC
  - Scheduler (Java-Programm)
- Auf den zu überwachenden Maschinen wird installiert
  - apache Webserver mit php- oder cgi-Funktionalität
  - ggf. SNMP
- Auf der zentralen Überwachungsmaschine (MASTER) wird installiert
  - mysql oder andere DB
  - apache
  - JRE
  - cUrl
  - php
  - JDBC
  - ZIPS-Schnittstelle (SMS/Mail)
  - Scheduler (Java-Programm)
  - GnuPlot

Plattformunabhängigkeit

Überall dort, wo ein Webserver mit php oder cgi bzw. SNMP installiert werden kann, ist eine Überwachung möglich.

Das heißt:

Windows, OS2, MAC, OS390, Solaris, SINIX, AIX, Router, Firewall .... sind überwachbar.

swest

Funktionsweise

- Definition der Überwachungsparameter

- Parametername (PARAM_NAME)
- Server, an den der Parameter gebunden ist (HOST)
- Über welche Collectoren wird der Server erreicht (WAY_TO_HOST)
- Start- und Endzeitpunkt der Messung
- Intervall der Messung
- Script, das den Wert liefert oder SNMP-MIB
- TCP/IP-Port, über den der Parameter auf dem Server abgefragt wird (PORT)
- Flag, ob der Parameter direkt aktiviert werden soll (ACTIVE)
- verscheidene Schwellwerte für die Alarmierung
- Wer soll wann wie alarmiert werden (EMail und/oder SMS)
- Flag, ob der Parameter bereits an den zu überwachenden Server verteilt wurde (ANNOUNCED)

... und ein paar mehr (z.B. Aufnahme in eine Langzeitstatistik) ....

swest

Funktionsweise

- Der Master

  - stellt eine Web-Oberfläche für die Überwachung/Administration zur Verfügung
  - schreibt die erfassten Parameter in die Datenbank
  - verteilt die Parameter an die zu überwachenden Server und gibt die Daten an die beteiligten Collectoren weiter
  - Sammelt auf den Collectoren angefallene Daten ein
  - Bereitet Statistiken grafisch auf
  - Führt Alarmierungen durch

Das Einsammeln von Messdaten erfolgt innerhalb des eingenen Netzsegmentes über einen Collector (Java-Programm + JDBC) und außerhalb des Netzsegmentes über „PHP/cU L".

Warum? Das reine Sammeln von Messdaten ist zeitunkritisch, weil die Timestamps der Messdaten mit übertragen weren. Der Collector muss aber sehr zeitnah Parameter ermitteln können (wenn z.B. das Intervall im Sekundenbereich liegt).

Funktionsweise

- Collector
  - Sammelt Messdaten von Servern innerhalb des eigenen Netzssegmentes in einer DB
  - Ermittelt die Daten über ein HTTP-Request oder über SNMP
  - Das Einsammeln der Daten erfolgt mit einem Java-Programm + JDBC

Funktionsweise

Erfassen eines Parameters auf einem Server, der sich in dem gleichen Netzsegment befindet, wie der Master

ANNOUNCED=FALSE	(default)
ACTIVE=TRUE	
WAY_TO_HOST=''	(gleiches Netzsegment)
HOST='servername'	(Server, der überwacht werden soll)
PORT=9001	(Port, über den der HTTP-Request erfolgen soll)
SCRIPTNAME='df.sh'	
PARAM1='/export/home'	

Funktionsweise

Eintragen des Scripts (ermitteln des freien Plattenplatzes)

```
#!/bin/ksh
export PATH=/bin:/usr/bin

df -k $1 | awk '{ if (NR==2) { split ($5,r,"%");print r[1];}
}'
```

Klar geht es mit SNMP einfacher – aber: SNMP ist nicht immer verfügbar.

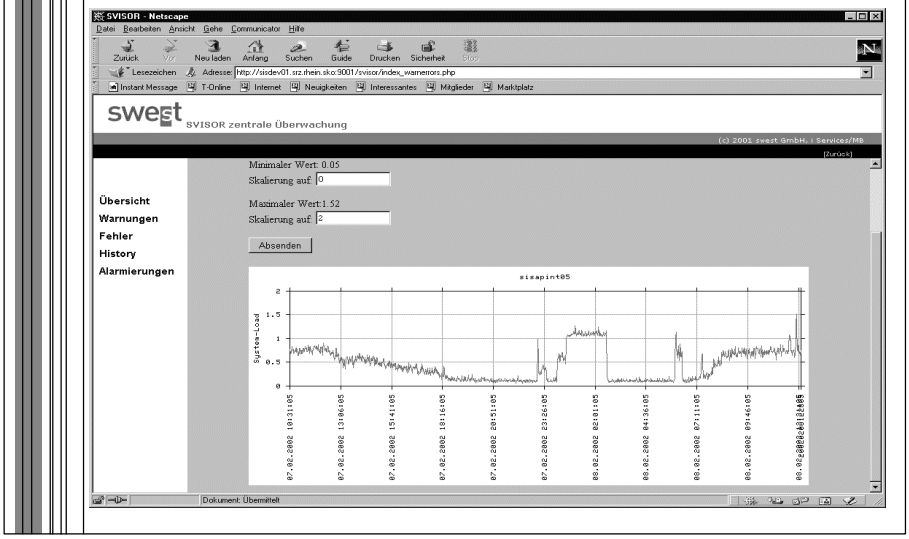
Funktionsweise

Nach Erfassen des Parameters:

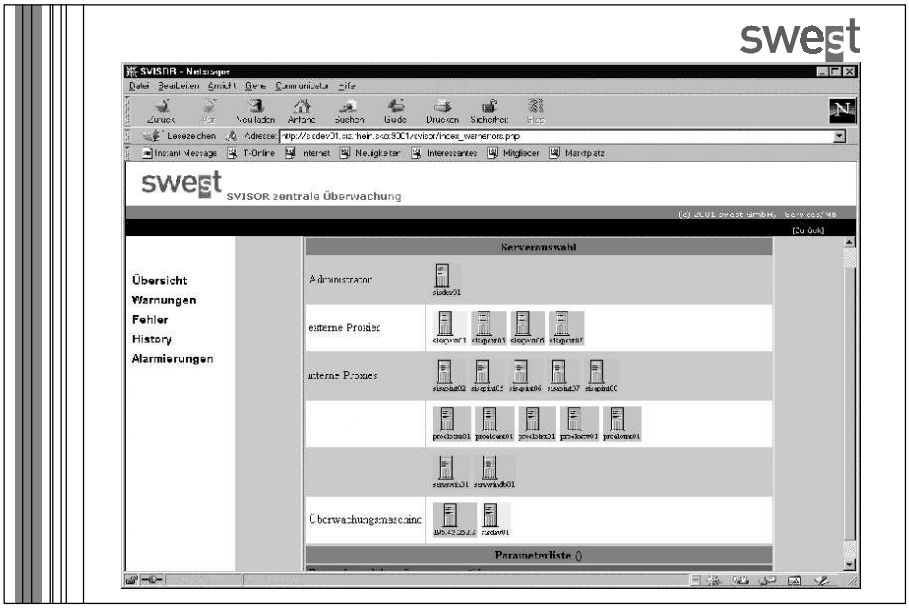
- wird dieser auf die Zielmaschine übertragen (per HTTP-Requenst über den angegebenen Port) Dabei wird eine PHP-Seite auf der Zielmaschiene erstellt, die das übertragene Script ausführen kann. Bei Erfolgreicher Übertragung des Parameters wird ANNOUNCED auf TRUE gesetzt.
- wird er in die Datenbank des Masters aufgenommen.

° Der Collector ruft in den angegebenen Zeitfenstern den Parameter per HTTP-Request ab und schreibt den Messwert in eine Tabelle.

° Der Master überprüft die gemessenen Werte mit den angegebenen Schwellwerten und alarmiert bei Bedarf.



Seite 19



Seite 20

Funktionsweise

Alarmierung

Eine SMS-Alarmierung kann über „http://www.sms-wap.com/" erfolgen. Ähnlich kann auch eine EMail-Alarmierung funktionieren. Im Haus der swest wird eine eigene Schnittstelle für die EMail- und SMS-Alarmierung bereitgestellt (ZIS).

Funktionsweise

Schön und gut – aber .....

Wie überwacht man einen Server hinter einer Firewall?

WAY_TO_HOST='col_fw1:9001'

Der Weg des zu überwachenden Servers geht hier über den Collector „col_fw1" über den Port 9001. Sicher: Dieser Port muss in der Firewall freigeschaltet werden.

Der Master trägt den Parameter in die Datenbank ein und überträgt das Script über den Collector „col_fw1" an die Zielmaschine.

Der Master sammelt nur Werte innerhalb der eigenen DMZ (WAY_TO_HOST=''), also nicht mehr die Parameter, der jetzt definiert wurde.

In der Datenbank des Collectors „col_fw1" steht der Parameter mit der Eintragung WAY_TO_HOST=''. Mit anderen Worten: Dieser Collector sammelt jetzt die Daten des hier definierten Parameters.

Funktionsweise

Schön und gut – aber .....

Wie überwacht man einen Server hinter mehreren Firewalls?

WAY_TO_HOST='col_fw1:9001,col_fw2:8800'

Der Weg des zu überwachenden Servers geht hier über den Collector „col_fw1" über den Port 9001 und über den Collector „col_fw2" über den Port 8800. Sicher: Der Port zwichen beiden Collectoren muss in der Firewall freigeschaltet werden.

Der Master trägt den Parameter in die Datenbank ein und überträgt das Script erst an den Collector „col_fw1". Dieser Collector trägt den Parameter in seine Datenbank mit WAY_TO_HOST='col_fw2:8800' ein. Danach erfolgt die Übertragung an „Col_fw2", der wiederum diesen Parameter mit dem Wert WAY_TO_HOST='' in die Datenbank einträgt und jetzt das Script an die Zielmaschine überträgt.

„Col_fw2" sammelt danach die Werte des Parameters in der eigenen Datenbank.

Funktionsweise

Wie bekommt der Master die Daten?

Auf dem Master wird in regelmäßigen Abständen die URL http://localhost:(Portnummer)/svisor/collect_all_measured_values.php aufgerufen. Dort wird die Datenbank nach allen Parametern durchsucht, die der Bedingung WAY_TO_HOST<>'' entsprechen.

TO_HOST=erster „Collector"
NEXT_HOSTS="nachfolgende Collectoren"
WAY_TO_HOST(alt)=TO_HOST+NEXT_HOSTS
WAY_TO_HOST(neu)=NEXT_HOSTS

Jetzt wird die URL http://TO_HOST:(Portnummer)/svisor/get_measured_values.php aufgerufen.
In diesem Script wird wie in „collect_all_measured_values.php" erfahren.

Der Vorgang wiederholt sich, bis WAY_TO_HOSTS='' ist. Dann liefert „get_measured_values.php" alle Werte an den Master zurück, die in der Datenbank gespeichert waren. Nach Übertragung der Daten werden diese aus der Datenbank von „col_fw2" entfernt.

Funktionsweise

Einen Blick hinter die Kulisse geworfen: (get_measured_values.php)

```
if (!strlen($WAY_TO_HOST)) {   // Wenn Zielkollektor erreicht
    $query="select TS,RETURN_VAL from T02_COLLECTED_VALUES where ....";
                $r=mysql_query($query);
                $nr=mysql_num_rows($r);
                print "MEASURED_DATA\n";
                print "$nr\n";                          // Hier steht, wieviele Werte kommen ....
                while ($o=mysql_fetch_object($r)) {
            print "$ID # $o->TS # $o->RETURN_VAL \n";
                }
   $query="delete from T02_COLLECTED_VALUES where T01_ID=$ID";
                mysql_query($query);
             } else {  // Weiterreichen an den naechsten Kollektor
                $pdp=strpos("$WAY_TO_HOST",",");
                $len=strlen("$WAY_TO_HOST");
                if (!$pdp) {
                        $pdp=$len;
                }
                $TO_HOST=substr("$WAY_TO_HOST",0,$pdp);
                $NEXT_HOSTS=substr("$WAY_TO_HOST",$pdp+1,$len-$pdp);
                $cf=curl_init();
                $pushvar['ID']=$ID;
                $pushvar['HOST']=$HOST;
                $pushvar['WAY_TO_HOST']=$NEXT_HOSTS;
                $d=curl_setopt
($cf,CURLOPT_URL,"http://$TO_HOST:9001/svisor/get_measured_values.php");
                .... ein paar Initialisierungswerte für cURL (heruasgeschnitten)
    $d=curl_exec($cf);
                print "$d";
                curl_close($cf);
            }
```

Seite 25

Funktionsweise

Wie unterscheidet sich:

collect_all_measured_values.php von get_measured_values.php?

collect_all_measured_values.php wird lokal vom Master aufgerufen und sieht in der Datenbank des Masters nach, welche Parameteter gesammelt werden müssen. Das Sammeln der Daten auf den Collectoren erfolgt mit get_measured_values.php.

Seite 26

swest

Beispiel

Der Endkunde kommt aus dem Internet an unsere DMZ1.
Dort steht der erste Proxy.
Durch eine Firewall werden die Datenpakete an den zweiten Proxy in der DMZ2 gesendet. Danach werden die Daten an einen dritten Proxy, der sich in der DMZ3 befindet übertragen, der natürlich auch durch eine Firewall getrennt ist. Schließlich werden die Daten an den Proxy in der DMZ4 gesendet und an das Ziel weitergereicht. Auch hier steht eine Firewall „im Weg".
Für die Administration der Anwendung stehen Pflegerechner in der KSK-Siegburg, der SK-Bonn, der SSK-Köln und der KSK-Köln, die über das CN an uns angebunden sind. Die Anwendung wird durch SOCKS-5-Proxies an das Ziel geroutet. Auch diese Proxies müssen überwacht werden, die über Firewalls voneinander getrennt sind.

swest

Vorschlag zur Bewertung

Folgende Fragen können über ein Überwachungssystem gestellt werden

- Welche Systemarchitekturen können überwacht werden (NT, OS/2, UNIX ....)? (nt, os/2, os390, aix, sinix, solaris ...)
- Wie lange dauert es, bis ein System in die Überwachung aufgenommen werden kann? (wenige Minuten)
- Ist eine Überwachung von Internetdiensten von „Außen" möglich, d.h. ohne Verwendung des swest-eigenen Netzwerkes? (ja)
- Können Systeme über mehrere Firewallebenen hinweg überwacht werden? (ja)
- Welcher Aufwand ist erforderlich, um Messparameter auf einem überwachten System zu verändern oder hinzuzufügen, welches sich hinter mehreren Firewalls befindet? (kann zentral per Browser erfolgen)
- Wie hoch wird das überwachte System zusätzlich belastet, wenn die Überwachungssoftware eingesetzt wird? (ca. 1%)
- Sind Prozesse notwendig, die auf der überwachten Maschine Administrationsrechte benötigen? (nein)

swest

Vorschlag zur Bewertung

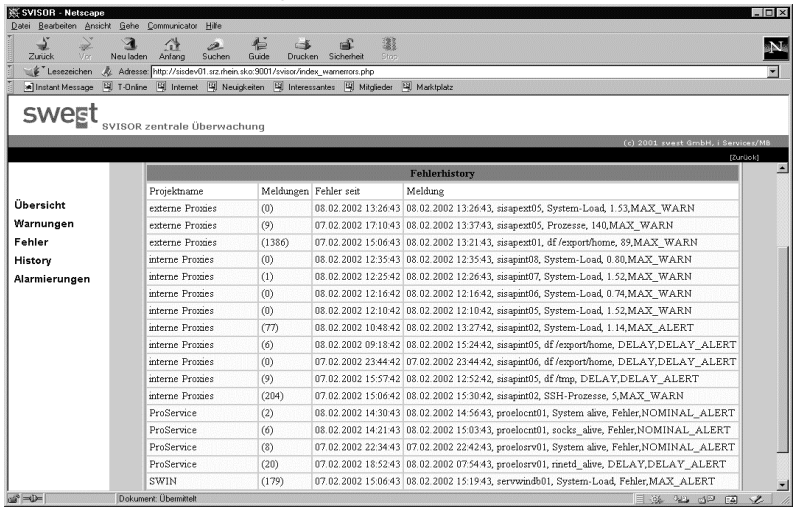
- Wieviele und was für Server sind für die Überwachung notwendig? (pro DMZ eine Netra X1, ca. 1600 - oder eine bereits existierende Maschine, die für die Überwachung verwendet werden kann
- Sind zusätzliche Softwarekomponenten auf den „Beobachtungsrechnern" erforderlich? (nein)
- Welche Software muss auf den zu überwachenden Systemen installiert werden? (reduzierter http-Server apache)
- Entstehen Lizenzkosten für die Überwachungssoftware? (nein, sie unterliegen alle der GPL und sind von Security freigeben worden - und die selbstentwickelten Programme sind swest-Eigentum)
- Ist das System Mandantenfähig - d.h. kann die Sicht auf überwachte Systeme bis auf Parameterebene benutzerabhängig eingestellt werden? (ja)

swest

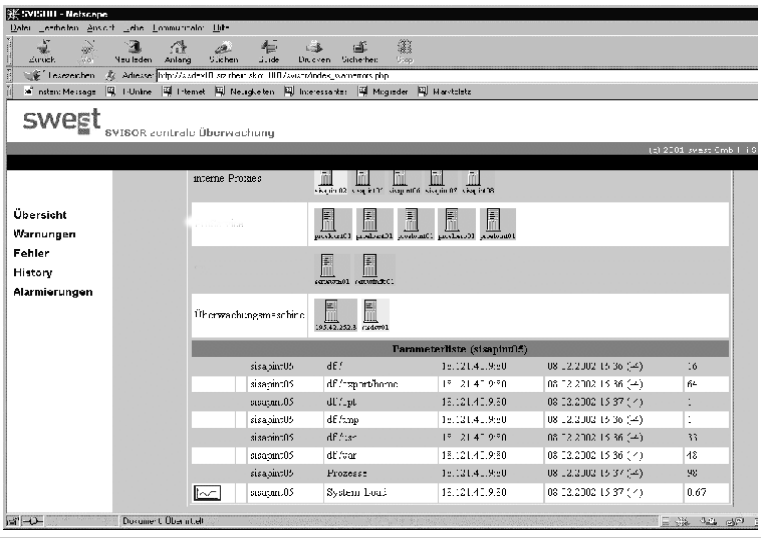
Ausblick

- ✗ Abfrage von Systeminformationen per Handy (lebt das System, wie sehen die Parameter aus?)
- ✗ Erweiterung um eine Softwareverteilungskomponente (kleinere Skripts lassen sich bereits verteilen)
- ✗ Bereitstellung eines Polling-Mechanismus zur Darstellung von Systeminformationen im Internet für Kunden.
- ✗ Erweiterung um eine Netzanalyse-Komponente (Laufzeitermittlung von Datenpaketen in de einzelnen Neztsegmenten).
- ✗ Verbesserung der grafischen Auswertung
  - Min/Max/Mittelwertdarstellung
  - Minuten/Stunden/Tages-/Nachtwerte
  - ....
- ✗ ...
- ✗ ...

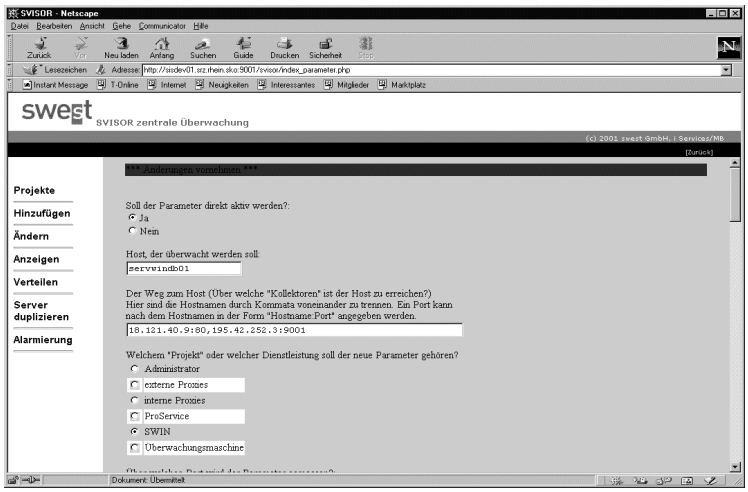
Fehlerhistory



Ein aktueller Parametersatz



Änderungen an Parametern



Herzlichen Dank für Ihr Interesse!!!