Von: Hans-Jürgen Barthen DL5DI [mailto:dl5di@gmx.de]
Datum: Mittwoch, 26. Mai 2010 20:12

Vorabversion

ircDDB Installationsinfo

Diese Dokumentation soll die Installation erklären, für alle diejenigen, die mehr sehen wissen als das, was unser Installations-Script automatisch erledigt.

Unsere Software benötigt auf dem Gateway eine Software, die die LastHeard-Informationen an eine Datenbank weiterleitet.

Wir haben im ersten Schritt dazu DstarMonitor von AE5PL eingesetzt.

XMonitor sollte diese Aufgabe mit leichten Anpassungen auch übernehmen können, uns stehen nur die Sourcen nicht zur Verfügung um es anzupassen.

Genauso könnten die LastHeard-Informationen (Rufzeichen und Einstiegs-Repeater-Modul mit Zeitstempel) auch aus anderen Applikationen bereitgestellt werden.

Eine Anpassung des Interfaces auf der Datenbankseite des ircDDB-Systems ist jederzeit möglich.

Wir gehen zunächst von dem aus, was wir haben, DSM.

DstarMonitor ist normalerweise unter /opt/dstarmon installiert.

Das ist nur wichtig, weil sich dstarmon und unser Tool den Postgres-Java-Connector teilen und der Pfad in dessen Config auch drinsteht, also ggf. angepasst werden müsste wenn man das anders installiert.

In den Properties für DSM muss ein Treiber hinzugefügt werden und zwar nur der zum Schreiben der MHeardTable.

Wie das im Detail geht steht in der Doku zur Software von Pete Loveall.

Diese Tabelle bekommt dann nur 3 Felder:

`target_cs, last_mod_time und arearp_cs`

XTrust-Benutzer sollten das Default-Reporting an dstarusers.org rausnehmen.

Einige XTrust-Systeme haben DstarMonitor bereits vorinstalliert, dort muss nur in den Properties eine Datenbank hinzugefügt werden.

In dem anhängenden Beispiel ist nur die Kommunikation an unseren Server drin, nicht nach dstaruser.org !

Die Datei sieht dann so aus:

```
stderr=/var/log/dstar/dstarmon.log
GWIntf=eth1
CtrlrIP=172.16.0.1
```

```
#
LHDriver=org.postgresql.Driver
LHURI=jdbc:postgresql://141.75.245.225:5432/allstarrouter
LHParameters=user;dm0px;password; 0aNliavqAAglLD1p
#
```

Anschließend folgen noch einige Einträge für xAPRS, die für uns keine Rolle spielen.

In den Pfad muss man dann noch den Postgres-Connector kopieren (postgresql-8.4-701.jdbc3.jar).

Gestartet wird DSM dann mit dem Script monstart.sh, wie es im Archiv enthalten ist.

Nun zum ircDDB-Client, der Software, die sich an den IRC-Server hängt und die dort kommenden Daten in die lokale Datenbank einarbeitet.

Wir legen den Benutzer „ircddb“ an mit Home-Directory „/opt/ircDDB“
Wir entpacken das Software-Archiv dorthin.

Das sollte dann so aussehen:

```
[root@db0myk-gw ircDDB]# ls -l
insgesamt 136
-rw-r--r-- 1 ircddb root 862 24. Mai 21:48 ircDDB.keystore
-rw-r--r-- 1 ircddb root 338 24. Mai 22:18 ircDDB.policy
-rw-r--r-- 1 ircddb root 318 23. Mai 23:16 ircDDB.properties
-rwxr-xr-x 1 ircddb root 890 25. Mai 18:53 run.sh
-rwxr-xr-x 1 ircddb root 51 25. Mai 19:23 start.sh
[root@db0myk-gw ircDDB]#
```

ircDDB ist ein allgemeines Projekt von Michael/DL1BFF (<http://github.com/dl1bff/ircDDB>).

Policy und keystroke sollen sicherstellen, dass beim Autoupdate niemand etwas falsches unterschiebt.

Run.sh ist ein endlos laufendes Script, was den autoupdate-Mechanismus beinhaltet.
Der Client kann per Kommando über IRC beendet werden, sichert dann die letzte lauffähige Version, sucht anschließend nach einem Update, wenn verfügbar installiert er es und startet neu.
Das lässt sich natürlich ändern, bzw. anpassen, falls ein eigener Client verwendet werden soll, der von woanders upgedatet werden muss.

Im IRC-Kanal kann man die Versionsnummer der Module aller Gateways abrufen (mit „/WHO *“).
Die Specs werden noch veröffentlicht.

Start.sh ist die Startdatei.

ircDDB wird wie alle anderen DStar-Komponenten vom Startscript /etc/init.d/dstar_gw gestartet.

Die Config-Datei ircDDB.properties sieht wie folgt aus:

```
irc_server_name=ircddb-server.mdx.de
```

```
irc_server_port=9007
irc_channel=#dstar

irc_password=*****
rptr_call=db0xyz

debug=0
rptr_fix_tables=yes
rptr_fix_unsync_gip=no

ext_app=de.mdx.ircDDBApps.RptrApp

jdbc_class=org.postgresql.Driver
jdbc_url=jdbc:postgresql://127.0.0.1:5432/dstar_global
jdbc_username=db-usrname
jdbc_password=db-passwd
```

Oben die Daten im ersten Block sind für den Zugang zum IRC-Server.

Die Zugangsdaten zum IRC-Server sind die gleichen wie die zur Datenbank in den DSM-Properties (s.o.).

Dann kommen 2 Schalter, die Unterschiede zwischen den Systemen ermöglichen:

```
rptr_fix_tables=yes bzw. no
    bei "yes" werden eventuell fehlende Zeilen in sync_gip und sync_mng
    angelegt. Vorsicht: die Voreinstellung ist "no", wir sollten bei unseren
    Repeatern also "rptr_fix_tables=yes" eintragen. Bei Test-Benutzern
    ausserhalb unseres "Cross-Trust-Netzes" laesst man die Zeile einfach weg
;)
```

Das heißt, wer keine sync_gip hat, lässt den Eintrag weg.

Wir haben den bei lcom- und OpenG2 gesetzt, beide nutzen die sync_gip zum Erreichen der anderen Gateways.

```
rptr_fix_unsync_gip=yes bzw. no
    dieser Parameter wird nur dann benoetigt, wenn die "rptr_fix_tables"-
    Funktion
    aktiviert ist. Beim Wert "yes" wird zusaetzlich zu einer neu angelegten
    Zeile in sync_gip auch eine Zeile in unsync_gip eingetragen.
    Bei openg2 gibt es diese Tabelle nicht, deswegen muss
    z.B. bei DB0FOX "rptr_fix_unsync_gip=no" drinstehen. Bei den anderen
    Relais kann die Zeile entfallen, weil die Voreinstellung "yes" ist.
```

Das sollte selbsterklärend sein.

Im unteren Block kommen dann noch die Zugangsdaten zur lokalen Datenbank.

Kein root-Zugriff wie beim gw_schedule-Script, sondern wie es sein sollte über den vorgesehenen User-Account.

In der run.sh findet man den Pfad zu dem Java-Postgres-Connector

```
CP="app.jar:ircDDB.jar:/opt/dstarmon/postgresql-8.4-701.jdbc3.jar"
```

Wir verwenden den unter dstarmon installierten Connector.

Zum Schluss dann noch was ganz Wichtiges, die Postgres-Trigger-Funktion.

Diese Funktion hängt sich direkt in Postgres rein, fängt INSERTs und UPDATEs vom Trust-Server ab und ersetzt die darin enthaltene arearp_cs-Info ggf. durch aktuellere Daten aus unserer Datenbank.

Installiert wird der Trigger als User postgres mit

```
psql -U dstar dstar_global < trigger.txt
```

Da Textfile befindet sich im Install-Directory.

73

Hans-Jürgen, DL5DI

SysOp @DA5UDJ, DB0LJ, DB0MYK

<http://www.prgm.org/>

<http://afuforum.prgm.org>

Tel Shack: +49 2652 938377

Skype: hjbarthen