Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles

Liebe Leserinnen und Leser,

gerade jetzt in der Urlaubszeit ist Sicherheit ein wichtiges Thema.
So wie Sie Einbrechern nicht auf Facebook verraten, dass das Haus nun drei Wochen ungestört geplündert werden kann, während Sie in Urlaub sind, so sollten Sie auch auf die Sicherheit der Daten achten. Verbindungen zwischen Server und Client, sowie der Datenaustausch zwischen Instanzen, sind die beliebtesten Einfalltore. Werden hier die Daten im Klartext übertragen, dann muss sich ein Spionageprogramm nicht einmal die Mühe machen, auf den Rechner zu gelangen.
Aus diesem Anlass haben wir uns in der aktuellen Ausgabe dem Thema der verschlüsselten Übertragung gewidmet.
Unsere Services unterstützen Sie gerne bei der Einrichtung sicherer Verbindungen.
Bleiben Sie sicher und geniessen Sie ungestört den Urlaub !

Viel Spaß mit den Tipps der aktuellen Ausgabe.
Ihr TechTeam

TechTipp: Informix Instanz mit SSL (Secure Sockets Layer) einrichten

Sicherheit wird immer wichtiger. Den Zugriff auf den Server und die Datenbankinstanz schützen meist Firewalls und strikte Passwortregeln. Ein häufiger Angriffspunkt ist jedoch die Kommunikation, bei der Daten den geschützten Bereich des Servers verlassen und über das Netzwerk transportiert werden. Netzwerkverkehr mitzulesen ist nicht schwer, daher empfiehlt es sich, dies durch Verschlüsselung zu unterbinden. Eine Möglichkeit, den Datentransfer abzusichern, ist die Verwendung des SSL-Protokolls.

SSL nutzt digitale Zertifikate für den Austausch von Verschlüsselungskeys und zur Authentifizierung. Die Zertifikate werden von einer „Certificate Authority (CA)“ vergeben und sind nur für einen begrenzten Zeitraum gültig. Steht kein offizielles Zertifikat einer CA zur Verfügung, so kann auch ein privates Zertifikat erstellt werden.

Der Austausch der Daten erfolgt bei SSL mit einem symmetrischen Verschlüsselungsalgorithmus. Ein unsymmetrischer Schlüssel (Public-Private-Key) wird für den Austausch des geheimen Key der symmetrischen Verschlüsselung genutzt.

Verbindet sich ein Client über eine SSL-Verbindung mit dem Server, so erfolgt ein Handshake, bei dem der Server sein digitales Zertifikat zum Client schickt.
Der Client prüft das Zertifikat, wofür der Client das digitale Zertifikat der CA besitzen muss. War dieser Handshake erfolgreich, so schickt der Client einen verschlüsselten symmetrischen Schlüssel an den Server, wozu er den asymmetrischen Key des Serverzertifikates nutzt. Der Server entschlüsselt den symmetrischen Key für die Datenübertragung durch sein Private Certificate.

Für die Dauer der Verbindung (so lange die Session existiert), kann nun die verschlüsselte Übertragung erfolgen.

Im folgenden Beispiel richten wir eine SSL-Verbindung zusätzlich zu den bestehenden Verbindungen über die TCP-Sockets und SharedMemory ein.

Grundvoraussetzung ist, dass das GSKit (Global Security Kit) installiert wurde. Dies wird (wenn es bei der Installation des Servers nicht explizit abgewählt wird), im Verzeichnis

$INFORMIXDIR/gskit

als Paket abgelegt.

Die Installation erfolgt als „root“ mittels „**./installgskit** “.
Je nach Version wird dieser Vorgang bereits bei der Installation von INFORMIX ausgeführt. Das Programm gsk8capicmd bzw. gsk8capicmd_64 wird dabei im Betriebssystem verankert. Auf Linux ist es unter /bin zu finden.
Ist die Installation erfolgt, so muss im nächsten Schritt eine KeyDB aufgebaut werden.

Dies geschieht als User „informix“. Man wechselt zuerst in das Verzeichnis **$INFORMIXDIR/ssl** und ruft dann folgende Befehle auf:

Erstellen der KeyDB:

```
gsk8capicmd_64  -keydb -create -db <instanzname>.kdb -pw <passwd>
-type cms -stash
```

Erstellen des Zertifikats:

```
gsk8capicmd_64  -cert -create -db <instanzname>.kdb -format ascii
-label <ssl_label> -pw <passwd> -dn "CN=<domain>" -size 1024
-default_cert yes
```

Statt eines neu erstellten Zertifikates kann auch ein bestehendes Zertifikat verwendet werden.
Die erstellten Dateien „<instanzname>.kdb“ und „<instanzname>.sth“ müssen im Verzeichnis $INFORMIXDIR/ssl stehen und die Rechte 600, sowie Owner und Group „informix“ besitzen.
Der Keystore enthält das digitale Zertifikat und die „root CA Zertifikate“ der anderen Server, mit denen die Instanz kommuniziert.
Jede Instanz hat ihren eigenen Keystore, wobei der Name mit „<instanzname>.kdb“ festgelegt ist. Die Datei „<instanzname>.sth“ beinhaltet das verschlüsselte Passwort des Keystore.

Sind die Voraussetzungen geschaffen, wird im nächsten Schritt die neue Verbindung mit einem neuen Namen (DBSERVERALIAS) und dem Protokoll „onsocssl“ in der Datei $INFORMIXSQLHOSTS eingetragen:

```
test1      onipcshm        172.16.41.230        dummy1
test1_soc  onsoctcp        172.16.41.230        9380
test1_ssl  onsocssl        172.16.41.230        9381
```

In der Datei $INFORMIXDIR/etc/$ONCONFIG wird der Name der Verbindung in die Liste der DBSERVERALIASES aufgenommen:

```
DBSERVERALIASES test1,test1_ssl
```

Das SSL_KEYSTORE_LABEL muss gesetzt werden, das beim Erstellen der Key-Datenbank vergeben wurde:

```
SSL_KEYSTORE_LABEL <ssl_label>
```

Die Parameter für die SSL-Verbindung können angegeben werden:

```
NETTYPE socssl,3,50,NET
```

Die Anzahl der Virtuellen Prozesse, die die Verschlüsselung durchführen, kann eingetragen werden (Default 1):

```
VPCLASS encrypt,num=3
```

Sind diese Schritte erfolgt, dann kann die Instanz gestartet werden.

Die SSL-Verbindung zeigt sich im „onstat -g ntt“:

```
Individual thread network information (times):
  netscb thread name sid open read write address
  …
  47447c98 socssllst 7 13:52:10 13:53:39 172.16.41.230|9381|socssl
```

Sowie im „onstat -g glo“:

```
Virtual processor summary:
 class        vps        usercpu    syscpu     total
 cpu          1          5.18       0.70       5.88
 aio          1          0.03       0.05       0.08
 lio          1          0.00       0.03       0.03
 pio          1          0.00       0.04       0.04
 adm          1          0.05       0.13       0.18
 soc          1          0.21       0.18       0.39
 msc          1          0.00       0.00       0.00
 encrypt      3          1.39       0.53       1.92
 ssl          3          1.61       2.01       3.62
 fifo         1          0.00       0.04       0.04
 total        14         8.47       3.71       12.18
```

Es ist empfehlenswert, mehr als einen VP der Klasse „encrypt“ zu starten. Wird dies nicht explizit in der $ONCONFIG mittels VPCLASS angegeben, so startet die Instanz mit der Aktivierung der SSL-Schnittstelle einen VP dieser Klasse.

Die Einrichtung einer DRDA-Verbindung mit SSL erfolgt analog. Das Protokoll lautet dann drsocssl, die restlichen Schritte sind gleich.

TechTipp: Informix Clients mit SSL (Secure Sockets Layer) einrichten

Informix Clients der folgenden Liste können über SSL mit dem Server kommunizieren:

- IBM® Data Server Driver für JDBC und SQLJ
- IBM Informix ESQL/C
- IBM Informix ODBC Driver
- DB-Access
- Enterprise Replication
- High-availability Data Replication (HDR) zwischen einem HDR Primary Server und einem oder mehreren Secondary Servern (HDR Secondary, SDS Secondary, or RSS Secondary)
- Distributed transaction über mehrere Datenbankserver hinweg
- dbexport, dbimport, dbschema, und dbload utility
- Connection Manager Verbindungen in einem Cluster

Am Client ist die Ablage der beiden Dateien für die Keystore Datenbank und das verschlüsselte Passwort nicht wie am Server auf $INFORMIXDIR/ssl festgelegt, sondern kann in der Datei $INFORMIXDIR/etc/conssl.cfg frei gewählt werden. Ist diese Datei nicht vorhanden, so werden die Dateien unter $INFORMIXDIR/etc/client.kdb und $INFORMIXDIR/etc/client.sth verwendet. Die Rechte auf beiden Dateien am Client sollten 666 sein, wobei dies nicht erzwungen wird.

Beispiel **conssl.cfg:**

```
SSL_KEYSTORE_FILE   /home/ifx/ssl_client.kdb
SSL_KEYSTORE_STH    /home/ifx/ssl_client.sth
```

Nun muss das Zertifikat vom Server auf den Client übertragen werden. Dazu wird dies am Server extrahiert mittels:

```
gsk8capicmd_64 -cert -extract -db <instanzname>.kdb -format ascii
-label <ssl_label> -pw <passwd> -target /tmp/forclient.cert
```

Am Client wird eine KeyDB erstellt mittels:

```
gsk8capicmd_64 -keydb -create -db <clientdb>.kdb -pw <passwd>
-type cms -stash
```

und das sicher übertragene Zertifikat (scp,sftp,...) hinzugefügt:

```
gsk8capicmd_64 -cert -add -db <clientdb>.kdb -pw <passwd> -label
<ssl_label> -file /tmp/forclient.cert -format ascii
```

TechTipp: HDR mit SSL (Secure Sockets Layer) einrichten

Um die Verbindung vom Primary Server zum Secondary Server auf SSL umzustellen reicht es, auf dem Secondary die selben Einträge für SSL in der $ONCONFIG vorzunehmen, wie auf dem Primary Server:

```
DBSERVERALIASES     test2,test2_ssl
NETTYPE             socssl,3,50,NET
SSL_KEYSTORE_LABEL  ssl_label
VPCLASS             encrypt,num=3,noage
```

Zudem muss die KeyDB vom Primary Server auf den Secondary Server übertragen und umbenannt werden. Im Beispiel haben wir den Primary „test1“ genannt, den Secondary „test2“. Die Datei $INFORMIXDIR/ssl/test1.kdb muss somit auf dem Secondary auf $INFORMIXDIR/ssl/test2.kdb kopiert werden, ebenso test1.sth auf test2.sth.

Ist dies erfolgt, so kann mit den bekannten Befehlen die HDR eingerichtet werden:
Am Secondary:

```
onmode -d secondary test1_ssl
```

Am Primary:

```
onmode -d primary test2_ssl
```

Der onstat -g dri zeigt die Verbindung:

```
Data Replication at 0x47372028:
Type    State  Paired server Last DR CKPT (id/pg) Supports Proxy Writes
primary on     test2_ssl     219 / 127            NA
```

Der „onstat -g ath“ zeigt die verschlüsselte Übertragung und die HDR:

```
Threads:
tid tcb      rstcb prty status                vp-class     name
…

  9 473a8950 0        1 running               12ssl*       socsslpoll
 10 473da568 0        1 running               13ssl*       socsslpoll
 11 473dad40 0        1 running               14ssl*       socsslpoll
 13 474c5178 0        2 sleeping forever       1cpu*       socssllst
…
111 62ee1958 465c80a8 3 cond wait smx pipe1     1cpu        smxsnd test2
112 63bc7028 465cfb28 3 cond wait netnorm       8encrypt*   smxrcv test2
113 63bc76a0 465cf268 1 sleeping secs: 1        1cpu        smxRecvSnd
114 63aec6b8 465d0ca8 3 cond wait netnorm       10encrypt*  smxrcv test2
115 63b9e568 465d1568 3 cond wait smx pipe1     1cpu        smxsnd test2
```

TechTipp: JDBC Clients mit SSL (Secure Sockets Layer) einrichten

JDBC nutzt, anders als z.B. esql/c und dbaccess, kein $INFORMIXDIR, so dass hier das Zertifikat mittels „$JAVA_HOME/bin/keytool“ importiert werden muss:

```
keytool -importcert -file /tmp/forclient.cert -keystore
$WORKDIR/etc/.keystore
```

Zusätzlich muss die SSL-Verbindung in den JDBC Properties eingetragen werden:

```
ic.db.sslConnection = true
javax.net.ssl.trustStore = WORKDIR/etc/.keystore
javax.net.ssl.trustStorePassword = <password>
```

Zum Abschluss der Artikel über SSL noch ein Performance Tipp aus der Praxis:
Nutzen Sie keine grössere „FET_BUF_SIZE“ mit SSL. Der Default brachte in Test die besten Ergebnisse beim Durchsatz.

TechTipp: Rolling Upgrade HDR/RSS/SDS

Seit der Version 12.10.FC5 ist für HDR, RSS und SDS ein „Rolling Upgrade“ unterstützt. Dies bedeutet eine deutlich reduzierte Downtime gegenüber den bisherigen Möglichkeiten der Migration, wie man sie bisher nur in einem CDR Cluster erreichen konnte (da die CDR die Replikation zwischen unterschiedlichen Versionen und Plattformen unterstützt).

Aktuell unterstützt der Rolling Upgrade nur die Migration von Version 12.10.FC4/12.10.FC4W1 auf 12.10.FC5.

Nicht möglich ist der Rolling Upgrade derzeit wenn:

- Ein Fixpack installiert werden muss, das eine Konvertierung erfordert.
- Upgrade auf 12.10.xC5 von 12.10.xC3, 12.10.xC2, or 12.10.xC1.
- Upgrade von älteren Versionn wie z.B. 11.50 und 11.70 auf 12.10.
- Upgrade von einem Patchport der 12.10.FC4X, ausser dies ist durch unseren IBM® Software Support empfohlen.

Die Migration im Rolling Upgrade erfolgt, indem nacheinander die Secondary Server Offline genommen werden und mit der neuen Informixversion wieder gestartet werden. Zu beachten ist, dass die unveränderte Konfigurationsdatei, die Datei sqlhosts, das Alarmprogramm, ggf. die Dateien der Remote Server und Remote Users, sowie eventuell Metadaten des Spatial Datablade im neuen Informixverzeichnis vorhanden sein müssen. um erfolgreich zu starten. Der Secondary Server meldet sich beim Primary Server an und die Replikation startet automatisch wieder an der Stelle, an der sie unterbrochen wurde.

Bei der Migration eines HDR Servers muss der Secondary zumindest mit NearSync arbeiten (also ohne Delay), damit die Migration erfolgen kann. Sollten nur RSS Secondary Server eingerichtet sein, empfiehlt es sich für die Migration einen der Server im Modus kurzfristig auf HDR Secondary zu ändern.

Ist ein Verbund verschiedener Replikationen im Einsatz, so sollte die Migration in folgender Reihenfolge durchgeführt werden:

1. Remote standalone (RS) secondary server
2. HDR secondary server
3. Shared disk (SD) secondary server
4. Primary server

Um am Ende auch den Primary Server zu migrieren, muss zuerst der HDR Secondary die Rolle des Primary Servers übernehmen. Dies erfolgt mittels
„onmode -d make primary <hdr_sec>“
am Secondary Server.

Ist die Software am Primary Server installiert, so wird dieser mittels „oninit -PHY“ gestartet. Anschliessend wird dieser Server mittels „onmode -d secondary <hdr_sec>“ als HDR Secondary gestartet. Der ehemalige Secondary Server <hdr_sec> hat ja inzwischen die Rolle des HDR Primary Servers eingenommen. Im Falle eines SDS Clusters wird der ehemalige Primary Server mittels „oninit -SDS“ gestartet, um als SDS Secondary aktiv zu werden.

Abschliessend kann der bisherige Primary Server wieder mittels „onmode -d make primary <prim>“ zum Primary werden, falls dies erforderlich ist. Bei gleich starken Servern im Cluster und dem Einsatz des ConnectionManagers ist dies jedoch meist nicht notwendig.

Sind Datablades im Einsatz, so werden automatisch auf dem migrierten Server Links von der alten auf die neue Version der Verzeichnisse der Datablades angelegt, da der Primary Server ja noch in den Prozeduren mit den alten Pfaden arbeitet, und dies an die Secondary Server überträgt:

```
TimeSeries.6.00.FC4 -> /opt/informix/extend/TimeSeries.6.00.FC6
TimeSeries.6.00.FC5 -> /opt/informix/extend/TimeSeries.6.00.FC6
```

Den aktuellen Zustand des Clusters können Sie mittels „onstat -g cluster“ abfragen, der im folgenden Artikel beschrieben wird.
Die Ausführliche Anleitung zum Rolling Upgrade finden Sie auch unter:
http://www.ibm.com/support/knowledgecenter/SSGU8G_12.1.0/com.ibm.mig.doc/ids_mig_290.htm

TechTipp: Optionen des ONSTAT (onstat -g cluster)

Der „onstat -g cluster“ zeigt alle Informix Instanzen an, die in einem Cluster eingebunden sind. Wird der Befehl auf dem Primary ausgeführt, so zeigt dieser Aufruf zuerst die Informationen des Primary zur aktuellen Position im Log, an der geschrieben wird.

Danach sind die weiteren Knoten im Cluster mit der Information zum Stand der Replikations aufgeführt. Unter „ACKed Log“ und „Applied Log“ ist zu sehen, welche Loginformation bereits vom jeweiligen Knoten bearbeitet, bzw. bestätigt wurde. Zudem sind je Knoten Informationen zum Status zu sehen. Diese beinhalten, ob der Knoten synchron oder asynchron arbeitet, verbunden und aktiv ist.

Beispiel:

```
onstat -g cluster

IBM Informix Dynamic Server Version 12.10.FC5W1 -- On-Line (Prim)

Primary Server:test1
Current Log Page:239,1615
Index page logging status: Enabled
Index page logging was enabled at: 2015/07/01 17:13:22

Server ACKed Log    Applied Log  Supports     Status
       (log, page)  (log, page)  Updates
test5  239,1615     239,1615     Yes    SYNC(SDS),Connected,Active
test2  239,1615     239,1615     No     ASYNC(HDR),Connected,On
test3  239,1615     239,1615     Yes    ASYNC(RSS),Connected,Active
```

Nicht im „onstat -g cluster“ sind die CDR-Server enthalten, die mir „cdr list server“ zu sehen sind:

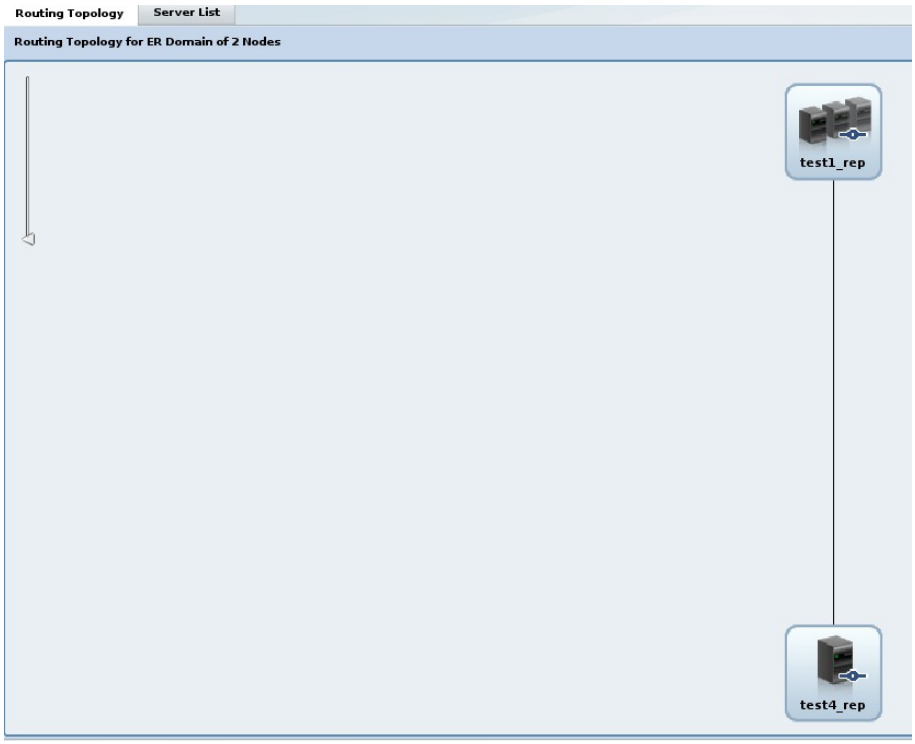
```
cdr list server

SERVER         ID STATE    STATUS      QUEUE  CONNECTION CHANGED
-------------------------------------------------------------------
test1_rep      1 Active   Local           0
test4_rep      4 Active   Connected       0 Jul  1 17:32:45
```

Wer diese Informationen in graphischer Form bevorzugt, der findet diese im OpenAdminTool unter „Replication → Cluster“, bzw. im Bereich „Replication → ER Domain“ für die Enterprise Replikation.

Die Ausgabe des „onstat -g cluster“ sieht im OAT so aus:

Die Ausgabe des „cdr list server“ zeigt sich im Bereich „Replication → ER Domain“:



Anmeldung / Abmeldung / Anmerkung

Der Newsletter wird ausschließlich an angemeldete Adressen verschickt. Die Anmeldung erfolgt, indem Sie eine Email mit dem Betreff „**ANMELDUNG**“ an **ifmxnews@de.ibm.com** senden.
Im Falle einer Abmeldung senden Sie „ABMELDUNG“ an diese Adresse.

Das Archiv der bisherigen Ausgaben finden Sie zum Beispiel unter:

http://www.iiug.org/intl/deu
http://www.iug.de/index.php?option=com_content&task=view&id=95&Itemid=149
http://www.informix-zone.com/informix-german-newsletter
http://www.drap.de/link/informix
http://www.nsi.de/informix/newsletter
http://www.cursor-distribution.de/index.php/aktuelles/informix-newsletter
http://www.listec.de/Newsletter/IBM-Informix-Newsletter/View-category.html
http://www.bereos.eu/software/informix/newsletter/

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit. Da uns weder Tippfehler noch Irrtümer fremd sind, bitten wir hier um Nachsicht falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

Die Autoren dieser Ausgabe

Gerd Kaluzinski IT-Specialist Informix Dynamic Server und DB2 UDB
IBM Software Group, Information Management
gerd.kaluzinski@de.ibm.com +49-175-228-1983

Martin Fuerderer IBM Informix Entwicklung, München
IBM Software Group, Information Management
martinfu@de.ibm.com

Markus Holzbauer IBM Informix Advanced Support
IBM Software Group, Information Management Support
holzbauer@de.ibm.com

Die Versionsinfo stammt aus dem Versions-Newsletter der CURSOR Software AG
http://www.cursor-distribution.de/download/informix-vinfo

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Gerd Kaluzinski (Redaktionsgarten)