Best practices in Firebird database security - from 2.5 to 4.0

Alex Peshkoff, Firebird
Alexey Kovyazin, IBSurgeon
Agenda

1) Authentication and Privileges
2) Best practices and demos
3) Mappings
1. Authentication and Privileges
Authentication and Privileges

- **Authentication** – “Confirm you are UserX”
- **Privileges** - “UserX is [not] allowed to do this”

- Firebird performs authentication on the server level, and grants privileges on server and database levels
  - By default any user with valid password to the server can establish connection to the database
  - To do something user requires privilege
Example: any user can connect!

```sql
isql -user SYSDBA -pass masterkey /:/d:\o30-etalon.fdb
Database: /:/d:\o30-etalon.fdb, User: SYSDBA
SQL> create user NEWUSER1 password '12345';
SQL> exit;

isql -user NEWUSER1 -pass 12345 /:/d:\o30-etalon.fdb
Database: /:/d:\o30-etalon.fdb, User: NEWUSER1
SQL> show database;
Database: /:/d:\o30-etalon.fdb
      Owner: SYSDBA
      PAGE_SIZE 8192
```
Example: ...any user can view basic metadata

SQL> show table AGENTS;
ID (DM_IDB) BIGINT Not Null Identity (by default)
NAME (DM_NAME) VARCHAR(80) CHARACTER SET UTF8 Nullable COLLATE NAME_COLL
IS_CUSTOMER (DM_SIGN) SMALLINT Nullable default 1 check(value in(-1, 1, 0))
IS_SUPPLIER (DM_SIGN) SMALLINT Nullable default 0 check(value in(-1, 1, 0))
IS_OUR_FIRM (DM_SIGN) SMALLINT Nullable default 0 check(value in(-1, 1, 0))
Example: ...but cannot access users data

SQL> select * from AGENTS;
Statement failed, SQLSTATE = 28000
no permission for SELECT access to TABLE AGENTS
Owner

- Owner of the database objects
  - Who created the object
- Owner of the database
  - Who created (= restored) database
- Owner has FULL access to the created objects
SYSDBA

• The ultimate ruler of the server
• SYSDBA has ALL privileges

• The popularity of SYSDBA based access is understandable, but not secure
Users

• Users are stored in security database
  • User names and hashes of passwords
  • Since Firebird 3 can be many security databases
• Authentication plugins (since Firebird 3)
  • Each plugin has own set of users
  • Yes, 2 SYSDBA with different passwords can co-exist!
Authentication plugins

• Auth plugins are dll/so libraries in folder plugins, and specified in firebird.conf:
  AuthServer = Srp, Legacy_Auth

• Each plugin can have own list of users and, optionally, UserManager
  UserManager=Srp, Legacy_UserManager
# Authentication plugins and User Manager plugins

<table>
<thead>
<tr>
<th>Authentication Plugin</th>
<th>UserManager Plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Srp</td>
<td>Srp</td>
</tr>
<tr>
<td>Srp256</td>
<td></td>
</tr>
<tr>
<td>Legacy_Auth</td>
<td>Legacy_UserManager</td>
</tr>
<tr>
<td>WinSSPI</td>
<td>-- gets data from Windows</td>
</tr>
<tr>
<td>Cluster (example in FB4 or HQbird)</td>
<td>– gets data from remote host</td>
</tr>
</tbody>
</table>
Example: identical users with different passwords

- CREATE USER newuser1 PASSWORD '12345' USING PLUGIN SRP;
- CREATE USER newuser1 PASSWORD '54321' USING PLUGIN Legacy_UserManager;
Example: identical users with different passwords

• `isql -user NEWUSER2 -pass 54321`
• Database: localhost:d:\o30-etalon.fdb, User: NEWUSER2
• SQL> exit;

• `isql -user NEWUSER2 -pass 12345`
• Database: localhost:d:\o30-etalon.fdb, User: NEWUSER2
• SQL> exit;
Order of plugins

- AuthServer = Srp, Legacy_Auth
  - AuthClient = Srp – changes the order

- By default, without specification, Firebird creates users using the first plugin defined in UserManager parameter:
  UserManager=Srp, Legacy_UserManager
- CREATE USER usr PASSWORD ‘12345’;
- Will be created with Srp
Authentication complete

- Let’s talk about privileges of authenticated user
What objects to be secured?

Security database

User NN

User NN

SYS DBA

Database 1

Stored Procedure

Generator

Trigger

Exception

View

Table 1

Record 1

Record 2

Record NN

Record 1

Record 2

Record NN

Database

Tables

Views

Stored Procs

Roles

Global actions

Create DB

Drop DB

Backup

....
Privileges

- Access to the objects
  - SELECT
  - INSERT, UPDATE, DELETE
  - EXECUTE
  - REFERENCES
- Managing privileges
  - GRANT
  - REVOKE
- Object management (DDL privileges)
  - CREATE
  - ALTER ANY
  - DROP ANY (ANY means access to non-owned objects)
GRANT/REVOKE

- **GRANT** <privileges>
  TO <user>| <role>| <object>
- **REVOKE** <privileges> FROM ...
  - Details are in Firebird Language Reference and Release Notes
2. Best practices for Firebird security
Out of the scope of Firebird security

- Protection of server and database
  - From the direct and embedded access
  - From stealing
- Consider encryption
  - Visit our “Firebird Database Encryption workshop” tomorrow!
Firebird security tuning can protect:

1) From non-authorized access from valid users
2) Brute force password attack
3) Access from unknown users
Don’t do, or useless SYSDBA role

- In Firebird 2.5, it is possible to create SYSDBA role (the same as SYSDBA), to protect from SYSDBA access
- Requires direct access to system tables
  - Does not work on Firebird 3!
- Weak protection
  - If attacker has physical access to database
Best practices: server level-1

1) Enable “Over the wire” encryption to prevent sniffing of users and password hashes

2) Use Srp256 and passwords 20+ symbols for users (SHA-256)

3) Use option to fetch passwords from files (isql, gbak, gfix):
   1) Parameter -FE(TCH)
   2) Easy change of passwords in all SQL scripts
Best practices: server level-2

1) Use separate security database for the database (can be set in databases.conf)

2) Create users with privileges to create new objects in database
   1) Use SQL scripts to [re-]create users
   2) Pseudo-Tables with List of Users to see the whole picture
Best practices: database level-1

• CREATE database with OWNER <> SYSDBA
• Define necessary ROLEs to manage all types of access
  • DDL privileges (GRANT CREATE/ALTER_ANY/DROP_ANY <object>\))
Best practices with ROLES-1

- Roles can be granted to other Roles!
  GRANT ROLE1 to BIGROLE

- Easy to define more granular access
  - ROLE1 covers operations with T1, T2, etc
  - ROLE2 covers operations with StoredProcedure1, SP2, etc
  - BIGROLE = ROLE1 AND ROLE
Best practices with ROLES-2

- **GRANT DEFAULT ROLE**
  - Once user is created, it immediately receives some default role:
    - Regular user
    - very restricted user
Record-level security

- At the moment the only method in community Firebird to implement record-level security is to use stored procedures
  - Access is granted inside the logic of the stored procedures
- Views
  - For simple cases
Stored procedures privileges

- Grant/revoke stored procedure to user
- Grant revoke/stored procedure to other stored procedure
- In Firebird 4, new feature
  - SQL SECURITY DEFINER
  - to grant all privileges to stored procedure
- Demo!
Why UDF are deprecated in 4.0?

- Potential attack with UDF code
- UDRs as replacement
  - UDR better protects from wrong parameters and other mistakes
- demo
3. Mappings: solution for external users
External users

• Some authentication plugins allow authentication of external users
  • External user = not stored in Firebird security database
• Execute statement on external (plugin Cluster)
• Server-wide plugins – WinSSPI
• Special plugins (for example, LDAP authentication)
Mappings

- Mappings create relationships between users of plugin and internal Firebird users and objects
- Can be global mappings and database-level mappings

- Plugin can send to the mapping any specific security object
  - E.g., WINSSPI transfers Windows groups and domain administrator attribute
Authentication block (simplified)

WinSSPI Plugin

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERNAME</td>
<td>Alex</td>
</tr>
<tr>
<td>GROUP</td>
<td>PowerUsers, Users</td>
</tr>
<tr>
<td>Adminstrator</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Example of mapping for auth block

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERNAME</td>
<td>Alex</td>
</tr>
<tr>
<td>GROUP</td>
<td>PowerUsers, Users</td>
</tr>
<tr>
<td>Adminstrator</td>
<td>Yes</td>
</tr>
</tbody>
</table>

CREATE MAPPING map1
USING PLUGIN WinSSPI
FROM USERNAME Alex to USER Donald

CREATE MAPPING map2
USING PLUGIN WinSSPI
FROM GROUP PowerUsers TO ROLE President;
Global Mapping

CREATE GLOBAL MAPPING
- adding GLOBAL makes it work for all databases which work with current security database
Example: Mapping between User and Role, Any user to User

CREATE MAPPING USR_CLUSTER9 USING PLUGIN CLUSTER FROM USER MUSER TO ROLE RDB$ADMIN;

CREATE MAPPING USR_CLUSTER_X USING PLUGIN CLUSTER FROM ANY USER TO USER MYUSER;
More security features in Firebird 4

- Built-in Cryptography functions (FB4)
- SET ROLE - change role without reconnect to the server
- DELETE USER
- System Privileges – to delegate several capabilities of SYSDBA/Owner to other users
  - CREATE(ALTER) ROLE SET SYSTEM PRIVILEGES TO USER_MANAGEMENT;
Example we promised: Disable connect from unwanted user

CREATE MAPPING deny_1 USING ANY PLUGIN FROM USER baduser TO ROLE BADROLE1;

CREATE MAPPING deny_2 USING ANY PLUGIN FROM USER baduser TO ROLE BADROLE2;
Summary

• Don’t use SYSDBA-only access
  • Try to avoid SYSDBA usage at all
• Use modern plugins (Srp256)
• Define privileges with ROLEs
• Use DEFAULT ROLE, etc
• Use stored procedures for fine tuning of the access
Thank you

• Questions?
• ak@ib-aid.com
• peshkoff@mail.ru