MySQL Security

John David Duncan
MySQL, AB

MySQL Users Conference
April 18-21, 2005
Some Security Priorities

- Your data is not stolen
- Your data is not tampered with
- Users can access only the data they are authorized for
- Users' passwords remain secure
- The database is not exploited to compromise your other servers and network
Multi-layer Security

DB
OS
Hardware
Building
Network
Topics

• Host-level security
• Network security
• Authentication in MySQL
• Authorization in MySQL
• The "MySQL Worm" of Jan. 27 '05
Rule 1:
Do not run the database server as a privileged system user.
Host Security

On Windows:

• Run a recent version of Windows (e.g. Win 2000, Win XP)

• Use NTFS
Network Security

- skip-networking: no TCP at all
- bind-address=127.0.0.1: only local TCP

- On Windows:
  - `mysql --protocol=pipe`
  - `mysql --protocol=memory`

- Use SSL
- Firewall
New Password Protocol

MySQL 4.1

• authentication exchange is less vulnerable to exploits
• password hash in mysql.user table grows from 16 chars to 41
• 4.1 clients can use old protocol
• if you need to upgrade the server without upgrading clients, use the --old-passwords option
SSL

• From slaves to master
• From clients to server

GRANT DELETE ON blog.* TO 'jdd'@'handcar' IDENTIFIED BY 'no-more-posts' REQUIRE SSL;

• REQUIRE SSL
• REQUIRE X509
• REQUIRE ISSUER
• REQUIRE SUBJECT
• REQUIRE CIPHER
Authentication & Authorization
MySQL User Names

- User & host: "jdd"@"handcar.lenoxway.net"
- Wildcard host: "jdd"@"%.lenoxway.net"
- Wildcard IP: "jdd"@"10.%"
- The anonymous user: "@'lenoxway.net'
- Netmask: ""@"198.144.209.0/255.255.255.240"
Web App User Names

- "update.php"@"10.4.8.:%"
- "newuser.php"@"10.4.8.:%"
- "subscribe.php"@"10.4.8.:%"

- "Subscription::New()"@"10.3.9.:%"
- "User::Login()"@"10.3.9.:%"

... allows script-specific or object-specific GRANTs ...

... good for SHOW PROCESSLIST ...
GRANT

- SELECT
- INSERT
- UPDATE
- DELETE
- SHOW DATABASES
  ... or --skip-show-databases
- etc...

mysql> help GRANT TYPES
GRANT ALL ON customer.*
TO 'francis'@'localhost'
IDENTIFIED BY 'black'
WITH MAX_CONNECTIONS_PER_HOUR 5
MAX_QUERIES_PER_HOUR 200
MAX_UPDATES_PER_HOUR 10 ;
Row-level security with SQL VIEWs

- CREATE VIEW sc_county as SELECT * FROM county WHERE state = 'SC' WITH CHECK OPTION

- GRANT SELECT, INSERT, UPDATE on sc_county to 'bob'@'handcar'

- Bob can select/insert/update the rows which meet the condition of the view
(use a GUI tool)
MySQL Worm

Jan 27, 2005

• scan TCP port 3306

• brute force password attack as user "root"

• CREATE TABLE bla (b blob)

• INSERT INTO bla VALUES(...executable code of worm ...)

• SELECT * FROM bla INTO DUMPFILE "app_result.dll"

• DROP TABLE bla
MySQL Worm

The UDF

- CREATE FUNCTION app_result SONAME 'app_result.dll'

- SELECT app_result(x)

- bot code in app_result.dll begins executing

- bot connects to an IRC channel, sets up an FTP server, etc.

- scan TCP port 3306...
MySQL 4.1.10a
...and 4.0.24, and 5.0.3

• UDFs must actually define all the functions of a correctly-written UDF or they will not be loaded

• --allow-suspicious-udfs option

• name of shared-object UDF file cannot contain pathname delimiter characters

• "root"@"%" and "@"%" accounts removed from the Windows distribution (other distributions never had them).

• CREATE TABLE will not overwrite existing files (".frm", ".MYI", etc.)