

A Developer Guide to

SQL Server Security

M. Choirul Amri (ASP.NET MVP)

choirul@mvps.org

<http://choirulamri.or.id>

<http://ilmukomputer.com>

Jakarta, April 30th, 2007

Session Agenda

- Security awareness for Developer
- SQL Injection Vulnerabilities
- SQL Injection Countermeasures
- Locking down SQL Server
- Encryption in SQL Server 2005

Security Awareness for Developer

- Infrastructure guys are not the only person who's responsible in security
- Security beyond the firewall:
 - Bad application design
 - Bad exception handling
 - Wrong deployment configuration
 - Wrong application logic

19 Programming Sins

- *95% of software bugs are caused by same 19 programming flaws... (Amit Yoran)*
 - Integer overflows
 - Buffer overruns
 - SQL Injection
 - Command injection
 - Cross site scripting
 - Format string problem
 - Etc...

Basic SQL Injection

- Replace string concatenation with another SQL script

```
SELECT COUNT(userid) FROM tblUser WHERE  
username = 'choirul' AND password = 'password'
```

- Replace 'choirul' with another string that always return true and add -- to comment any remaining SQL statements, example:

```
' OR 2=2 --
```

One step ahead:

Querying Metadata and modify data

- Attacker querying database metadata from sysobjects

`select id, name,o from sysobjects WHERE xtype = 'U'—`

- Combine with the existing statement with UNION keyword

`m;' UNION select id, name,o from sysobjects WHERE xtype = 'U'--`

- Modify data

`m;' UPDATE Customers SET companyname='guest' where city = 'elgin'--`

Dangerous: Executing xp_cmdshell

- Access to command prompt from SQL command
- Delete some files:
`m;' exec master..xp_cmdshell 'del d:\pleasedelete.txt'--`
- Stop SQL Server 😊
`m;' exec master..xp_cmdshell 'net stop MSSQL$SQL2K5'—`

Or..

Delete your database file!

DEMO

SQL Injection Attack

SQL Injection: Countermeasures

- Never use user input as a string concatenation element
- Avoid string concatenation in SQL statement
- Use Parameter collection of ADO.NET Command object for parameter parsing
- Use stored procedure as possible
- Validate user input, never trust them
- Client side validation is not enough (java script can be removed)
- Utilize built in database constraint
- Change sa password
- Remove built in administrator

Demo

SQL Injection Countermeasures

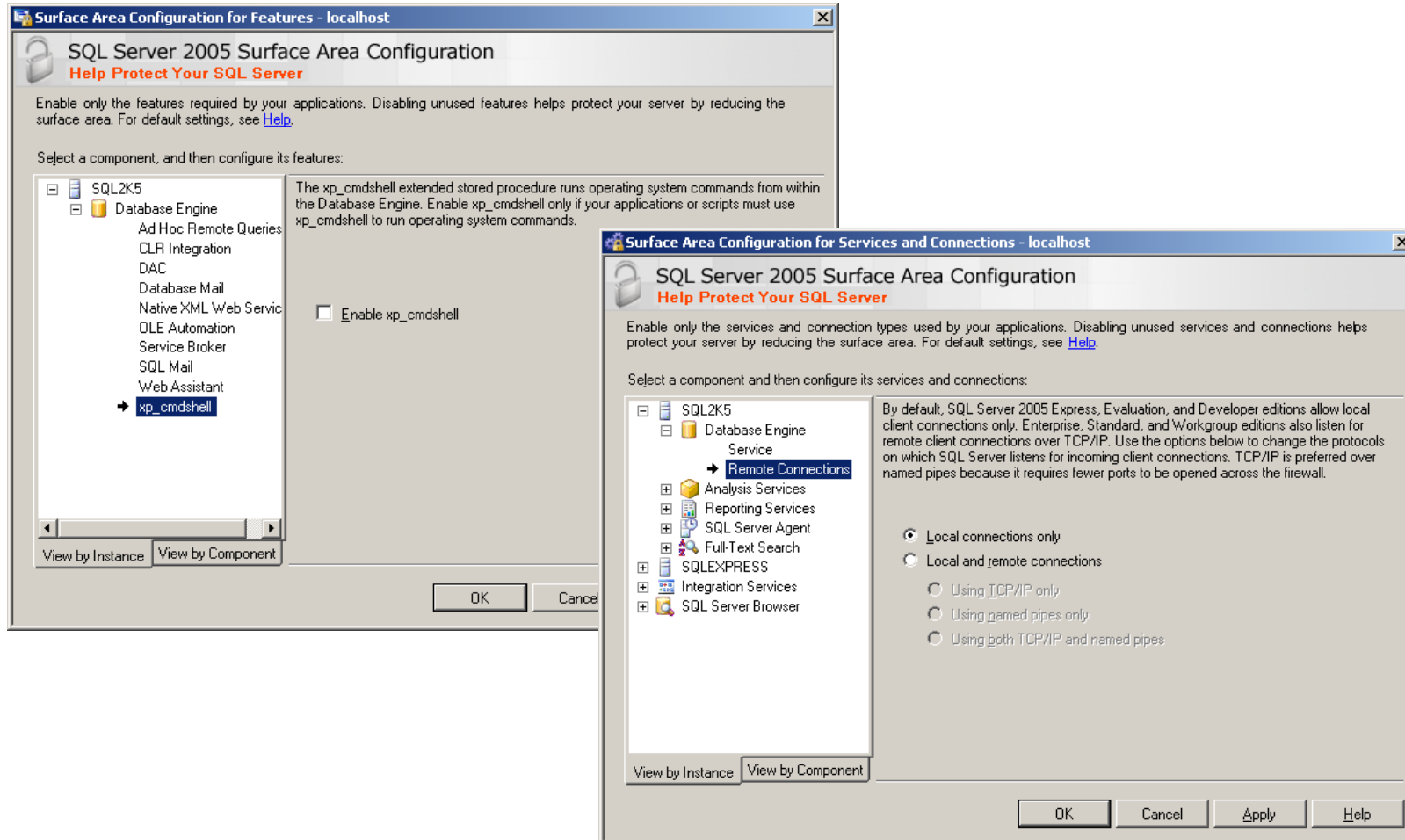
Demo

Bypassing Client Side Validation

Locking Down SQL Server

- A developer perspectives:
 - Disable xp_cmdshell (default ON in SQL2000)
 - Disable OPENROWSET and OPENDATASOURCE (default ON in SQL2000)
 - Don't enable SQL CLR if not needed
 - Minimize protocol used
 - In SQL 2005: Use surface Area configuration
 - Disable remote access if not needed
 - All setting can be modified using sp_configure
 - SQL authentication: create user mapping, don't use the real SQL user
 - Use Application Role if possible

Surface Area Configuration SQL Server 2005



Use sp_configure to lock down SQL Server

The screenshot shows the Microsoft SQL Server Management Studio interface. The query window contains the following SQL code:

```
1
2 --Enable advanced option
3 exec sp_configure 'show advanced option', 1
4 reconfigure
5
6 --locking down sql server: disable unneeded setting
7 exec sp_configure 'xp_cmdshell', 0
8 exec sp_configure 'Ad Hoc Distributed Queries', 0
9 reconfigure
10
```

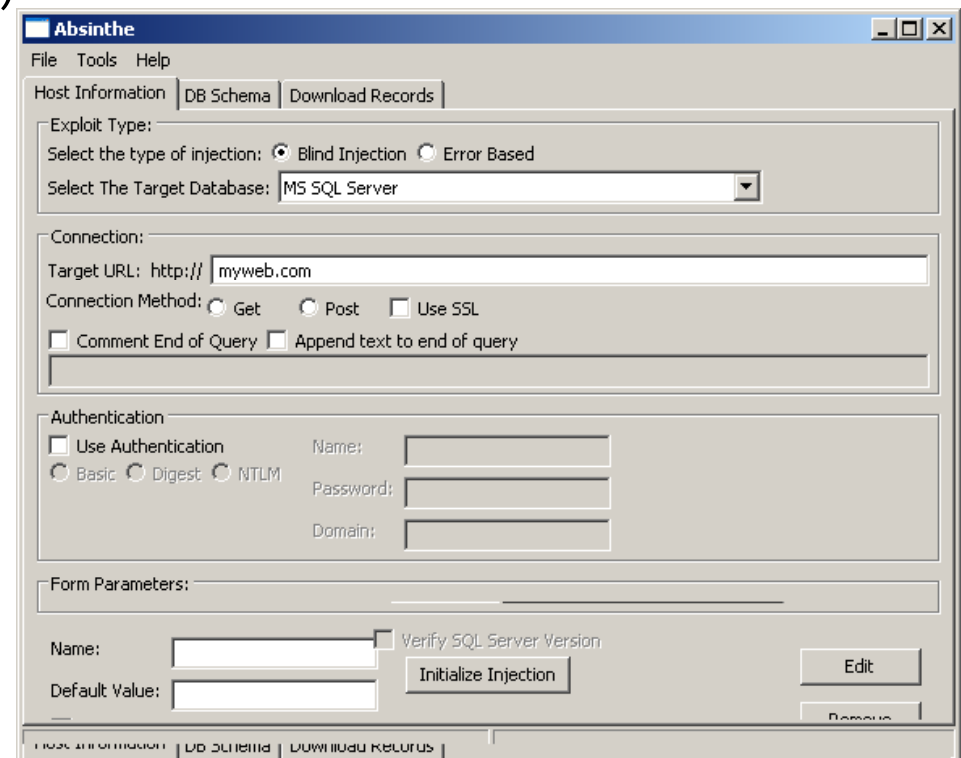
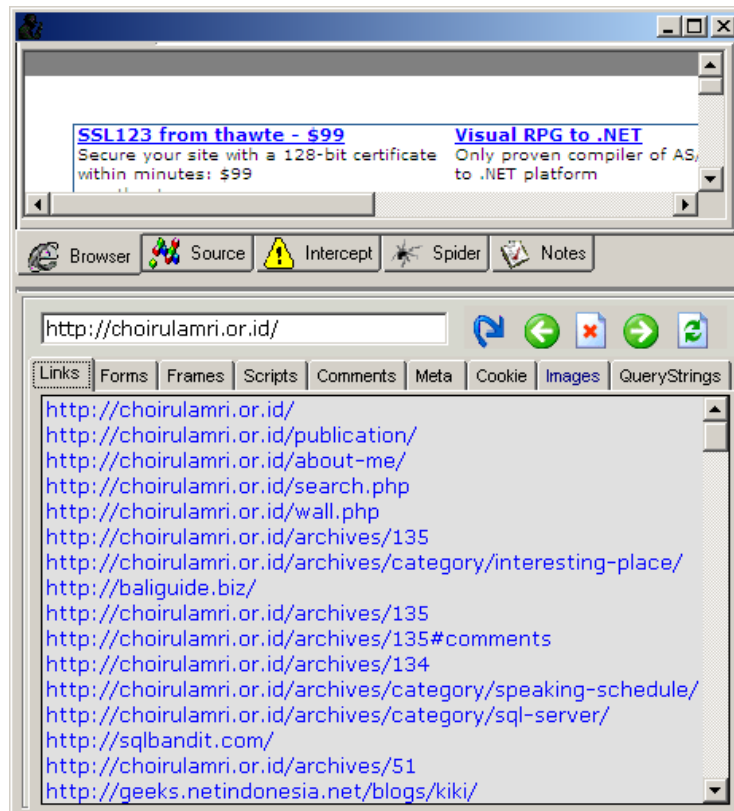
The Results window displays the following table:

	name	minimum	maximum	config_value	run_value
52	server trigger recursion	0	1	1	1
53	set working set size	0	1	0	0
54	show advanced options	0	1	1	1
55	SMD and DMO XPs	0	1	1	1
56	SQL Mail XPs	0	1	0	0
57	transform noise words	0	1	0	0
58	two digit year cutoff	1753	9999	2049	2049
59	user connections	0	32767	0	0
60	user options	0	32767	0	0
61	Web Assistant Procedures	0	1	0	0
62	xp_cmdshell	0	1	0	0

The status bar at the bottom indicates: Query executed successfully. matrix\sql2k5 (9.0 RTM) MATRIX\mca (59) master 00:00:00 62 rows

Security Tool

- Absinthe (SQL Injection Scanner)
- Sleuth (Web Analyzer)



SQL Server Encryption

- SQL 2000:
 - Use tools made by Michael Pole
 - Download and follow tutorials in SQLServerCentral.com
- SQL 2005
 - Support built in encryption
 - Symmetric and asymmetric key

Further References

- Morris Lewis, "SQL Server Security Distilled" (Apress)
- Michael Howard, David LeBlanc, John Viega, "19 Deadly Sins of Software Security" (McGraw-Hill Osborne Media)
- Michael Howard and David LeBlanc, "Writing Secure Code" (Microsoft Press)