Database Security
SQL Server 2012

Russ McMahon – Associate Prof of Information Technology – CECH – UC

1 of only 21 ABET accredited programs in the US

Jasmine Hix – Senior IT (Software Dev) – Helped with the slides

http://cech.uc.edu/it/current-students.html

IT Expo -- The 2014 IT Expo will be on April 15th in the Rec Center. Be sure to come by to see the seniors & their projects!

#hottestcollegeinAmerica
60 Years of Computing in the Greater Cincinnati Area

March 29, 1953 “Cincinnati goes on the map scientifically in May” – Jack Dudley (Enquirer Financial Editor)

IBM 701 – No 6/19

Monthly rental: $17,000 or $147,337.79 in today's terms

GE Louisville received its computer (Univac) in 1954 and was used in the Payroll and other related applications. This was one of the first non-scientific applications of computing.
If I can’t answer your question, then...
OWASP Top 10

- **A1 Injection**
- **A2 Broken Authentication and Session Management**
  - Insider or external attacker gains access to the system’s password database. User passwords are not properly hashed, exposing every users’ password to the attacker.
- **A5 Security Misconfiguration**
- **A6 Sensitive Data Exposure**
- **A9 Using Components with Known Vulnerabilities**
  - Slammer virus, Jan 25, 2003
  - 75000 infections within 10 min
  - Patch has been out for 6 months
A1 Injection

- Option #1: Use of Prepared Statements (Parameterized Queries)
- Option #2: Use of Stored Procedures
- Option #3: Escaping all User Supplied Input
- Also Enforce: Least Privilege
- Also Perform: White List Input Validation
- Consider the business value of the lost data.
- What is your legal liability if this data is exposed?
- See Microsoft Technet article on SQL Injection
- See OWASP SQL Injection Prevention Cheat Sheet
A5 Security Misconfiguration

- Is any of your software out of date?
- Are any unnecessary features enabled or installed?
- Are default accounts and their passwords still enabled and unchanged?
- Does your error handling reveal stack traces or other overly informative error messages to users?
- Are the security settings in your development frameworks and libraries not set to secure values?

Example: App server comes with sample applications that are not removed from your production server. Said sample applications have well known security flaws attackers can use to compromise your server.
Too Much Information

Error Executing Database Query.

[Macromedia][Oracle JDBC Driver]No more data available to read.

The error occurred in D:\wwwroot\pd\pd.cfm: line 13
11 : </head>
12 :
13 : <cfquery name="datetime" datasource="slick">
14 : SELECT sysdate curdate
15 : FROM dual

Server Error in '/' Application.

An error has occurred while establishing a connection to the server. When connecting to SQL Server 2005, this failure may be caused by the fact that under the default settings SQL Server does not allow remote connections. (provider: Named Pipes Provider, error: 40 - Could not open a connection to SQL Server)

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: An error has occurred while establishing a connection to the server. When connecting to SQL Server 2005, this failure may be caused by the fact that under the default settings SQL Server does not allow remote connections. (provider: Named Pipes Provider, error: 40 - Could not open a connection to SQL Server)

Source Error:

An unhandled exception was generated during the execution of the current web request. Information regarding the origin and location of the exception can be identified using the exception stack trace below.

Stack Trace:

[SqlException (0x80131904): An error has occurred while establishing a connection to the server. When connecting to SQL Server 2005, this failure may be caused by the fact that under the default settings SQL Server does not allow remote connections. (provider: Named Pipes Provider, error: 40 - Could not open a connection to SQL Server)]
A6 Sensitive Data Exposure

- Scenario: An application encrypts credit card numbers in a database using automatic database encryption. However, this means it also decrypts this data automatically when retrieved, allowing an SQL injection flaw to retrieve credit card numbers in clear text. The system should have encrypted the credit card numbers using a public key, and only allowed back-end applications to decrypt them with the private key.

- Scenario: The password database uses unsalted hashes to store everyone’s passwords. A file upload flaw allows an attacker to retrieve the password file. All of the unsalted hashes can be exposed with a rainbow table of precalculated hashes.

- Don’t store sensitive data unnecessarily. Discard it as soon as possible. Data you don’t have can’t be stolen.
  - Of course, it helps if you know what data you really have and what data you really don’t have.
  - Some internal department/individual that is keeping sensitive information.
New In SQL Server 2012

- Multiple ways to access the server
- New Role Management
- New permissions
- Default Schema for Groups
- Audit Enhancements
- Transparent Data Encryption
**Principals and Securables**

- As with prior editions of SQL Server, all security is centered around principals, which are the logins and users who are granted permission to access database objects and securables, which are the objects that you can limit access to.
  - Users, database roles, application roles
  - Tables, views, procedures (are schema scoped)

- Grant permissions at the schema level.

- Encapsulate access through modules
  - Stored procedures and user-defined functions.

- Users that are not mapped to logins provide an alternative to using application roles.

- SQL Server does not yet implement row-level security as part of the database engine.
  - SQL Server Label Security Toolkit (CodePlex)
  - Limited cell-level encryption – must use VARBINARY
Access Control

- Use administrator privileges only when needed.
- Minimize the number of administrators.
- Have multiple distinct administrators if more than one is needed.
- Confer trust selectively.
- Avoid dependency on the built-in \administrators Windows group.
- Use user-defined server roles as an alternative to assigning server-level privileges to individual users.
- If you are an ISP, have distinct owners for databases; not all databases should be owned by sa.
- Leave the Cross-Database Ownership Chaining setting off unless multiple databases are deployed at a single unit.
- Migrate usage to selective trust instead of using the TRUSTWORTHY property.
Multiple Ways to Access the Server

Before:
- There was a single way to access a database.
- A login principal was defined that allowed a principal to access the server using Windows credentials, where the login was then mapped to a user within the database for access.

Contained Database Authentication (CDB):
- A db that is isolated from other DBs and from the instance of SQL Server that hosts the db.
- This makes it easier to have user accounts that are limited to a single db.
- Allows user authentication to be performed by the database, reducing the database dependency on the logins of the instance of SQL Server.
- Authentication data is stored in the user db
- Users can’t perform db-level operations (only DML)
- Eliminates problems with orphaned/unused logins
- DB becomes more portable

It is a best practice to use only Windows logins whenever possible.
- Using Windows logins with SQL Server achieves single sign-on and simplifies login administration.
- Use login triggers for more granular control of the login process.
New Role Management

- User-defined server roles are
  - Applied at the server level
  - Prevents unnecessary use of the sysadmin role
  - Available to allow certain users rights to VIEW SERVER STATE and VIEW ANY DATABASE.
  - To manage user-defined server roles use CREATE SERVER ROLE, ALTER SERVER ROLE, and DROP SERVER ROLE.
  - IS_ROLEMEMBER is added to check the membership of database roles.

- Best practices for db object mgmt
  - Encapsulate access within modules.
  - Manage permissions via database roles or Windows groups.
  - Use permission granularity to implement the principle of least privilege.
  - Do not enable guest access in any database except MSDB.
  - Use users without logins instead of application roles.
New Permissions

- New GRANT, DENY, and REVOKE permissions to:
  - CONTROL/VIEW DEFINITION/TAKE OWNERSHIP/REFERENCES/ALTER ON a search property list are available.
  - ALTER ANY SERVER ROLE, CREATE SERVER ROLE, and CONTROL/VIEW DEFINITION/TAKE OWNERSHIP/ALTER ON a server role.
  - ALTER ANY AVAILABILITY GROUP, CREATE AVAILABILITY GROUP, and CONTROL/VIEW DEFINITION/TAKE OWNERSHIP/ALTER ON an availability group.
Logins & Guest User

- A login only can only be granted authorization to objects in a database if a database user has been mapped to the login.

- A special user, guest, exists to permit access to a database for logins that are not mapped to a specific database user.
  - Because any login can use the database through the guest user, it is suggested that the guest user not be enabled
  - Except in the MSDB database

- Users that are not mapped to logins provide an alternative to using application roles.
  - Can invoke selective impersonation by using the EXECUTE AS statement and allow that user only the privileges needed to perform a specific task.
Default Schema for Groups

- Unlike previous versions of SQL Server, you can now define a default schema for a Windows group. When an object is created by a Windows user and when a default schema is not specified, SQL Server no longer automatically creates a schema.

- A schema is simply a named container for database objects.
  - Each schema is a scope that fits into the hierarchy between database level and object level, and each schema has a specific owner.
  - The owner of a schema can be a user, a database role, or an application role.
  - Schemas solve an administration problem that occurs when each database object is named after the user who creates it.
Best Practices for Using Schemas

- Group like objects together into the same schema.
- Manage database object security by using ownership and permissions at the schema level.
- Have distinct owners for schemas or use a user without a login as a schema owner.
- Not all schemas should be owned by dbo.
- Minimize the number of owners for each schema.
- For users mapped to Windows groups, try and limit each Windows user to one Windows group that has database access.
- Use two-part names for database object creation and access.
Audit Is More Resilient to Failures

- HIPAA, PCI DSS, etc. compliance

Before:
- Write failures may silently lose Audit records
- Used ON_FAILURE = SHUTDOWN

Now:
- Automatically recover from most file or network errors
- New “ON_FAILURE = FAIL_OPERATION”
  - The FAIL_OPERATION option for the ON_FAILURE event, fails an operation that would otherwise make an audit event to be written to a failed audit target.
- A new WHERE clause allows SQL Server Audit the ability to filter audit events before they are written to the audit log (record filtering)
- New “MAX_FILES” option to allow customers to control the amount of audit information collected without losing audit records.
- Use SQL Server Audit instead of SQL Trace
SQL Server 2012 has built-in data encryption, both at a cell level and encryption of an entire database (TDE) via DDL (Enterprise version only).

This will encrypt the data and log files during I/O so that you don’t have to change your code, but anytime the data is at rest, it will stay encrypted (including when it is backed up).

- Does not encrypt data stored using SQL Server’s FILESTREAM storage feature.
  - BitLocker encrypts the entire volume, which would include FILESTREAM storage and non-SQL Server database files.

Requires secure encryption keys and key management. A key management hierarchy is built into SQL Server.

- DPAPI (Data Protection API)
- In SQL Server 2012, the service and database master keys use the AES_256 encryption algorithm
- TDE supports asymmetric keys that are provisioned by Extensible-Key Mgmt (EKM) in which keys can be managed by an external source
Data Encryption

- Encrypt high-value and sensitive data.
- Use symmetric keys to encrypt data, and asymmetric keys or certificates to protect the symmetric keys.
- Password-protect keys and remove master key encryption for the most secure configuration.
- Do not delete pre-provisioned system certificates in the master database.
- Always back up the service master key, database master keys, and certificates by using the key-specific DDL statements.
- Always back up your database to back up your symmetric and asymmetric keys.
- TDE is recommended for encrypting existing applications or for performance sensitive applications.
- Cell-level encryption can be used for defense in depth both for a database encrypted by TDE and for limited access control through the use of passwords.
- Use EKM with both database-level and cell-level encryption for more comprehensive key management and hardware-based cryptography.
System Stored Procedures

- Disable `xp_cmdshell` unless it is absolutely needed.
- Disable COM components once all COM components have been converted to SQLCLR (Common Language Runtime).
- Disable both mail procedures (Database Mail and SQL Mail) unless you need to send mail from SQL Server. Prefer Database Mail as soon as you can convert to it.
- Use Policy-Based Management to enforce a standard policy for extended procedure usage.
- Document each exception to the standard policy.
- Do not remove the system stored procedures by dropping them. Can cause problems when applying service packs.
- Do not modify the default permissions on system objects.
- Do not DENY all users/administrators access to the extended procedures.
Resources

- SQL Server 2012 Security Best Practices - Operational and Administrative Tasks (White paper); Bob Beauchemin, Jan, 2012
- SQL Server 2012 Security Enhancements, youtube video
- Introducing SQL Server 2012 Security and Auditing Improvements, channel0.msdn.com
- SQL Server Security & Compliance
  - Fewest number of vulnerabilities (National Vulnerability DB)
- SQLSecurity.com
- SQL Server 2012 Keeps Your Data a Little More Secure by Don Kiely
- OWASP.org