

Inhaltsverzeichnis

Danksagung.....	21
Die Autoren der Beiträge	23
Chefautor und Fachgutachter.....	26
 1. Was ist SQL-Injection?.....	 27
1.1 Einführung	27
1.2 Wie funktionieren Webanwendungen?.....	28
1.2.1 Eine einfache Anwendungsarchitektur.....	30
1.2.2 Eine kompliziertere Architektur.....	31
1.3 Wie funktioniert SQL-Injection?	32
1.3.1 Aufsehererregende Beispiele	36
1.4 Wie kann das passieren?.....	40
1.4.1 Dynamische Stringerstellung	40
1.4.2 Falsche Behandlung von numerischen Typen	43
1.4.3 Falsche Behandlung von Mehrfachübertragungen	48
1.4.4 Unsichere Datenbankkonfiguration	50
1.5 Zusammenfassung.....	52
1.6 Schneller Überblick.....	53
1.6.1 Wie funktionieren Webanwendungen?.....	53
1.6.2 Wie funktioniert SQL-Injection?	53
1.6.3 Wie kann das passieren?.....	54
1.7 Häufig gestellte Fragen	54

2. Anwendungen auf Anfälligkeit für SQL-Injection prüfen.....	57
2.1 Einführung	57
2.2 Anfälligkeiten für SQL-Injection finden.....	58
2.2.1 Deduktives Testen.....	59
2.2.2 Datenbankfehler	69
2.2.3 Antworten der Anwendung	84
2.2.4 Blinde Injektion.....	90
2.3 Das Vorhandensein von Schwachstellen bestätigen	95
2.3.1 Zahlen und Strings unterscheiden	95
2.3.2 Inline-SQL-Injection	96
2.3.3 Die SQL-Injection abschließen	105
2.3.4 Zeitverzögerungen	116
2.4 Das Aufspüren von Schwachstellen automatisieren.....	118
2.4.1 Werkzeuge zum automatischen Aufspüren von Schwachstellen	119
2.5 Zusammenfassung.....	129
2.6 Schneller Überblick.....	130
2.6.1 Anfälligkeiten für SQL-Injection finden.....	130
2.6.2 Das Vorhandensein von Schwachstellen bestätigen	131
2.6.3 Das Aufspüren von Schwachstellen automatisieren.....	131
2.7 Häufig gestellte Fragen	132
 3. Quellcode untersuchen	 135
3.1 Einführung	135
3.2 Quellcode auf Anfälligkeiten für SQL-Injection untersuchen	136
3.2.1 Gefährliche Programmiertechniken.....	138
3.2.2 Gefährliche Funktionen.....	146
3.2.3 Daten verfolgen	150
3.2.4 Code von Android-Anwendungen untersuchen	159
3.2.5 PL/SQL- und T-SQL-Code überprüfen	166

3.3	Automatisierte Quellcodeprüfung.....	173
3.3.1	Graudit.....	175
3.3.2	Yet Another Source Code Analyzer (YASCA)	176
3.3.3	Pixy	176
3.3.4	AppCodeScan	177
3.3.5	OWASP LAPSE+.....	178
3.3.6	Microsoft Source Code Analyzer for SQL Injection.....	179
3.3.7	Microsoft Code Analysis Tool .NET (CAT.NET)	179
3.3.8	RIPS – Statischer Quellcodeanalysator für Schwachstellen in PHP-Skripten	180
3.3.9	CodePro AnalytiX	180
3.3.10	Teachable Static Analysis Workbench	181
3.3.11	Kommerzielle Tools zur Quellcodeuntersuchung.....	181
3.3.12	Fortify	182
3.3.13	Rational AppScan Source Edition	184
3.3.14	CodeSecure.....	184
3.3.15	Klocwork Solo	185
3.4	Zusammenfassung.....	185
3.5	Schneller Überblick.....	186
3.5.1	Quellcode auf Anfälligkeiten für SQL-Injection untersuchen	186
3.5.2	Automatisierte Quellcodeprüfung.....	187
3.6	Häufig gestellte Fragen	187
4.	Anfälligkeiten für SQL-Injection ausnutzen	191
4.1	Einführung	191
4.2	Gebräuchliche Exploit-Techniken	193
4.2.1	Stapelabfragen	195
4.3	Das Datenbanksystem ermitteln	196
4.3.1	Identifizierung mit Rückmeldungen	197
4.3.2	Blinde Identifizierung.....	202

4.4	Daten mithilfe von UNION-Anweisungen abrufen	204
4.4.1	Übereinstimmende Spalten	205
4.4.2	Übereinstimmende Datentypen	208
4.5	Bedingte Anweisungen	213
4.5.1	Variante 1: Zeitgestützte Vorgehensweise	214
4.5.2	Variante 2: Fehlergestützte Vorgehensweise.....	219
4.5.3	Variante 3: Inhaltsgestützte Vorgehensweise.....	221
4.5.4	Strings	222
4.5.5	Die Technik erweitern.....	224
4.5.6	Fehlermeldungen zur SQL-Injection nutzen.....	225
4.5.7	Fehlermeldungen in Oracle	228
4.6	Das Datenbankschema auflisten.....	232
4.6.1	SQL Server	233
4.6.2	MySQL.....	238
4.6.3	PostgreSQL	243
4.6.4	Oracle	244
4.7	Injektion in INSERT-Abfragen	248
4.7.1	Szenario 1: Vom Benutzer ausgewählte Daten einfügen.....	249
4.7.2	Szenario 2: INSERT-Fehler auslösen	252
4.7.3	Weitere Szenarien	255
4.8	Rechte erhöhen.....	255
4.8.1	SQL Server	256
4.8.2	Oracle	263
4.8.3	Das Erfordernis für CREATE PROCEDURE überwinden	266
4.9	Passworthashes stehlen	268
4.9.1	SQL Server	268
4.9.2	MySQL.....	270
4.9.3	PostgreSQL	271
4.9.4	Oracle	272
4.10	Out-of-band-Kommunikation	276

4.10.1	E-Mail.....	277
4.10.2	HTTP/DNS.....	281
4.10.3	Dateisystem.....	282
4.11	SQL-Injection auf Mobilgeräten	286
4.12	SQL-Injektionsangriffe automatisieren.....	291
4.12.1	Sqlmap	291
4.12.2	Bobcat	293
4.12.3	BSQL.....	294
4.12.4	Weitere Programme	296
4.13	Zusammenfassung.....	297
4.14	Schneller Überblick.....	298
4.14.1	Gebräuchliche Exploit-Techniken	298
4.14.2	Das Datenbanksystem ermitteln	299
4.14.3	Daten mithilfe von UNION-Anweisungen abrufen	299
4.14.4	Bedingte Anweisungen	299
4.14.5	Das Datenbankschema auflisten.....	300
4.14.6	Injektion in INSERT-Abfragen	300
4.14.7	Rechte erhöhen.....	300
4.14.8	Passworthashes stehlen	301
4.14.9	Out-of-band-Kommunikation	301
4.14.10	SQL-Injection auf Mobilgeräten	301
4.14.11	SQL-Injektionsangriffe automatisieren.....	301
4.15	Häufig gestellte Fragen	302

5. Blinde SQL-Injection..... 303

5.1	Einführung	303
5.2	Anfälligkeiten für blinde SQL-Injection finden und bestätigen	305
5.2.1	Allgemeine Fehlermeldungen provozieren	305
5.2.2	Abfragen mit Nebenwirkungen einschleusen	306

5.2.3	Aufteilen und Ausgleichen	306
5.2.4	Häufige Situationen bei der blinden SQL-Injektion	309
5.2.5	Techniken zur blinden SQL-Injektion	310
5.3	Zeitgestützte Techniken	322
5.3.1	Datenbankabfragen verzögern	323
5.3.2	Überlegungen zur zeitgestützten Deduktion	334
5.4	Antwortgestützte Techniken	334
5.4.1	Antworttechniken in MySQL	335
5.4.2	Antworttechniken in PostgreSQL	337
5.4.3	Antworttechniken in SQL Server	338
5.4.4	Antworttechniken in Oracle	341
5.4.5	Mehr als ein Bit an Informationen abrufen	342
5.5	Alternative Kanäle	345
5.5.1	Datenbankverbindungen	346
5.5.2	Datenschleusung über DNS	348
5.5.3	Datenschleusung per E-Mail	353
5.5.4	Datenschleusung über HTTP	354
5.5.5	Datenschleusung durch ICMP	357
5.6	Automatisieren der blinden SQL-Injektion	357
5.6.1	Absinthe	357
5.6.2	BSQL Hacker	359
5.6.3	SQLBrute	362
5.6.4	Sqlmap	364
5.6.5	Sqlninja	366
5.6.6	Squeeza	368
5.7	Zusammenfassung	369
5.8	Schneller Überblick	370
5.8.1	Anfälligkeiten für blinde SQL-Injection finden und bestätigen	370
5.8.2	Zeitgestützte Techniken	371
5.8.3	Antwortgestützte Techniken	371

5.8.4	Alternative Kanäle	371
5.8.5	Automatisieren der blinden SQL-Injektion	372
5.9	Häufig gestellte Fragen	372
6.	Das Betriebssystem angreifen.....	375
6.1	Einführung	375
6.2	Zugriff auf das Dateisystem	377
6.2.1	Dateien lesen	377
6.2.2	Dateien schreiben	395
6.3	Befehle des Betriebssystems ausführen	407
6.3.1	MySQL.....	408
6.3.2	Microsoft SQL Server	409
6.3.3	Oracle	412
6.3.4	PostgreSQL	421
6.4	Den Zugriff sichern.....	423
6.5	Zusammenfassung.....	426
6.6	Schneller Überblick.....	426
6.6.1	Zugriff auf das Dateisystem	426
6.6.2	Befehle des Betriebssystems ausführen	427
6.6.3	Den Zugriff sichern.....	427
6.7	Quellen	428
6.8	Häufig gestellte Fragen	428
7.	Themen für Fortgeschrittene	431
7.1	Einführung	431
7.2	Eingabefilter umgehen.....	432
7.2.1	Groß- und Kleinschreibung	432
7.2.2	SQL-Kommentare	433

7.2.3	URL-Codierung.....	434
7.2.4	Dynamische Ausführung von Abfragen	437
7.2.5	NULL-Bytes	438
7.2.6	Verschachtelung von Ausdrücken, die entfernt werden.....	439
7.2.7	Abschneiden von Zeichen ausnutzen	439
7.2.8	Anwendungseigene Filter umgehen	442
7.2.9	Nicht standardmäßige Eintrittspunkte	443
7.3	SQL-Injection zweiter Ordnung	445
7.3.1	Anfälligkeiten für SQL-Injection zweiter Ordnung finden.....	448
7.4	Clientseitige SQL-Injektion.....	451
7.4.1	Zugriff auf lokale Datenbanken	451
7.4.2	Clientseitige Datenbanken angreifen	452
7.5	Kombinierte Angriffe	454
7.5.1	Erbeutete Daten nutzen	454
7.5.2	Cross-Site Scripting	455
7.5.3	Betriebssystembefehle in Oracle ausführen.....	456
7.5.4	Schwachstellen in Bereichen für authentifizierte Benutzer ausnutzen.....	456
7.6	Zusammenfassung.....	458
7.7	Schneller Überblick.....	459
7.7.1	Eingabefilter umgehen	459
7.7.2	SQL-Injection zweiter Ordnung	459
7.7.3	Clientseitige SQL-Injektion.....	460
7.7.4	Kombinierte Angriffe	460
7.8	Häufig gestellte Fragen	461

8. Schutzmaßnahmen auf Codeebene **463**

8.1	Einführung	463
8.2	Domain-Driven Security	464
8.3	Parametrisierte Anweisungen	470

8.3.1	Parametrisierte Anweisungen in Java	471
8.3.2	Parametrisierte Anweisungen in .NET (C#)	473
8.3.3	Parametrisierte Anweisungen in PHP	475
8.3.4	Parametrisierte Anweisungen in PL/SQL	476
8.3.5	Parametrisierte Anweisungen in Mobilanwendungen	477
8.3.6	Parametrisierte Anweisungen in HTML5	478
8.4	Eingaben validieren	479
8.4.1	Whitelists	479
8.4.2	Blacklists	484
8.4.3	Eingabevalidierung in Java	485
8.4.4	Eingabevalidierung in .NET	486
8.4.5	Eingabevalidierung in PHP	487
8.4.6	Eingabevalidierung in Mobilanwendungen	488
8.4.7	Eingabevalidierung in HTML5	488
8.5	Ausgaben codieren	488
8.5.1	Codierung bei der Übermittlung an die Datenbank	489
8.5.2	NoSQL-Injection verhindern	498
8.6	Kanonisierung	499
8.6.1	Vorgehensweisen zur Kanonisierung	500
8.7	Designtechniken zur Vermeidung der Gefahren von SQL-Injektion	502
8.7.1	Gespeicherte Prozeduren verwenden	502
8.7.2	Abstraktionsschichten verwenden	504
8.7.3	Umgang mit sensiblen Daten	505
8.7.4	Offensichtliche Objektnamen vermeiden	506
8.7.5	Datenbank-Honeypots einrichten	507
8.7.6	Weitere Quellen zur sicheren Entwicklung	508
8.8	Zusammenfassung	509
8.9	Schneller Überblick	510
8.9.1	Domain-Driven Security	510
8.9.2	Parametrisierte Anweisungen	510

8.9.3	Eingaben validieren	510
8.9.4	Ausgaben codieren	511
8.9.5	Kanonisierung	511
8.9.6	Designtechniken zur Vermeidung der Gefahren von SQL-Injektion.....	511
8.10	Häufig gestellte Fragen	512

9. Schutzmaßnahmen auf Plattformebene 515

9.1	Einführung	515
9.2	Laufzeitschutz.....	516
9.2.1	Firewalls von Webanwendungen	517
9.2.2	Abfangfilter	524
9.2.3	Bearbeitbare und nicht bearbeitbare Eingaben.....	530
9.2.4	Strategien auf URL- und Seitenebene	531
9.2.5	Aspektorientierte Programmierung (AOP)	532
9.2.6	Intrusion-Detection-Systeme (IDS)	533
9.2.7	Datenbankfirewall.....	533
9.3	Die Datenbank absichern	534
9.3.1	Den Zugriff auf die Anwendungsdaten einschränken	534
9.3.2	Den Zugriff auf den Datenbankserver einschränken	540
9.4	Weitere Überlegungen zur Bereitstellung	543
9.4.1	Unnötige Informationen unterdrücken	543
9.4.2	Ausführlichere Webserverprotokolle.....	549
9.4.3	Den Web- und den Datenbankserver auf getrennten Computern unterbringen	549
9.4.4	Netzwerkzugriffssteuerung einrichten	550
9.5	Zusammenfassung.....	550
9.6	Schneller Überblick.....	551
9.6.1	Laufzeitschutz.....	551
9.6.2	Die Datenbank absichern	551

9.6.3	Weitere Überlegungen zur Bereitstellung	551
9.7	Häufig gestellte Fragen	552

10.	SQL-Injektionsangriffe erkennen und sich von den Auswirkungen erholen	555
10.1	Einführung	555
10.2	Einen vermuteten SQL-Injektionsangriff untersuchen	556
10.2.1	Sinnvolle Ermittlungspraktiken	556
10.2.2	Digitale Artefakte analysieren	559
10.3	Was tun nach einem Angriff?	588
10.3.1	Die Auswirkungen eindämmen	588
10.3.2	Die betroffenen Daten ermitteln.....	589
10.3.3	Zuständige Personen informieren	590
10.3.4	Die Aktionen des Angreifers auf dem System ermitteln.....	591
10.3.5	Erholung von einem SQL-Injektionsangriff	593
10.4	Zusammenfassung.....	599
10.5	Schneller Überblick.....	599
10.5.1	Einen vermuteten SQL-Injektionsangriff untersuchen	599
10.5.2	Sinnvolle Ermittlungspraktiken	599
10.5.3	Digitale Artefakte analysieren	600
10.5.4	Durch SQL-Injektionsangriffe verursachte Aktivitäten erkennen	600
10.5.5	Das Vorliegen eines erfolgreichen Angriffs bestätigen	601
10.5.6	Die Auswirkungen eindämmen	601
10.5.7	Die betroffenen Daten ermitteln.....	601
10.5.8	Zuständige Personen informieren	601
10.5.9	Die Aktionen des Angreifers auf dem System ermitteln.....	601
10.5.10	Die Payload eines Angriffs bestimmen	602
10.5.11	Erholung von einem SQL-Injektionsangriff	602
10.6	Häufig gestellte Fragen	602

11. Referenz	605
11.1 Einführung	605
11.2 Grundlagen von SQL	606
11.2.1 SQL-Abfragen.....	606
11.2.2 Die Menge der Ergebnisse einschränken	612
11.3 Schnellreferenz zur SQL-Injection	613
11.3.1 Anfälligkeiten für SQL-Injection aufspüren.....	614
11.3.2 Die Datenbankplattform ermitteln	618
11.3.3 Spickzettel für Microsoft SQL Server.....	622
11.3.4 Spickzettel für MySQL.....	632
11.3.5 Spickzettel für Oracle.....	636
11.3.6 Spickzettel für PostgreSQL.....	642
11.4 Filter zur Eingabevalidierung umgehen	645
11.4.1 Filter für Anführungszeichen	645
11.4.2 HTTP-Codierung.....	647
11.5 Fehlerbehebung bei SQL-Injektionsangriffen	648
11.6 SQL-Injection auf anderen Plattformen	651
11.6.1 Spickzettel für DB2	651
11.6.2 Spickzettel für #.....	652
11.6.3 Spickzettel für Ingres	654
11.6.4 Spickzettel für Sybase.....	655
11.6.5 Microsoft Access.....	657
11.7 Literatur	657
11.7.1 Whitepaper zur SQL-Injection	657
11.7.2 Spickzettel zur SQL-Injection	657
11.7.3 SQL-Injektionswerkzeuge	658
11.7.4 Passwortknacker	659
11.8 Schneller Überblick.....	659
11.8.1 Grundlagen von SQL	659

11.8.2 Schnellreferenz zur SQL-Injection 660

11.8.3 Filter zur Eingabevalidierung umgehen 660

11.8.4 Fehlerbehebung bei SQL-Injektionsangriffen 660

11.8.5 SQL-Injection auf anderen Plattformen661

Stichwortverzeichnis 663