



SQL Smuggling The Attack That Wasn't There

OWASP

Israel 2007
December 3rd

Avi Douglén
Senior AppSec Consultant
Comsec Global
Douglén@hotmail.com

Based on

http://www.ComsecGlobal.com/Research/SQL_Smuggling.pdf



Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation

<http://www.owasp.org>



Agenda

- SQL Injection Revisited
- Classic Smuggling
- Introducing SQL Smuggling
- Common SQL Smuggling
- Unicode Smuggling
- Applicability
- Recommendations and Conclusions



OWASP SQL Injection Revisited

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

SQL Injection Basics

- Well known attack against DB
- Main cause: Lack of data validation
- Causes input to “break out” of query
- Most often based on special characters
 - ▶ E.g. Quote (') to terminate strings
- Rest of string seen as SQL commands

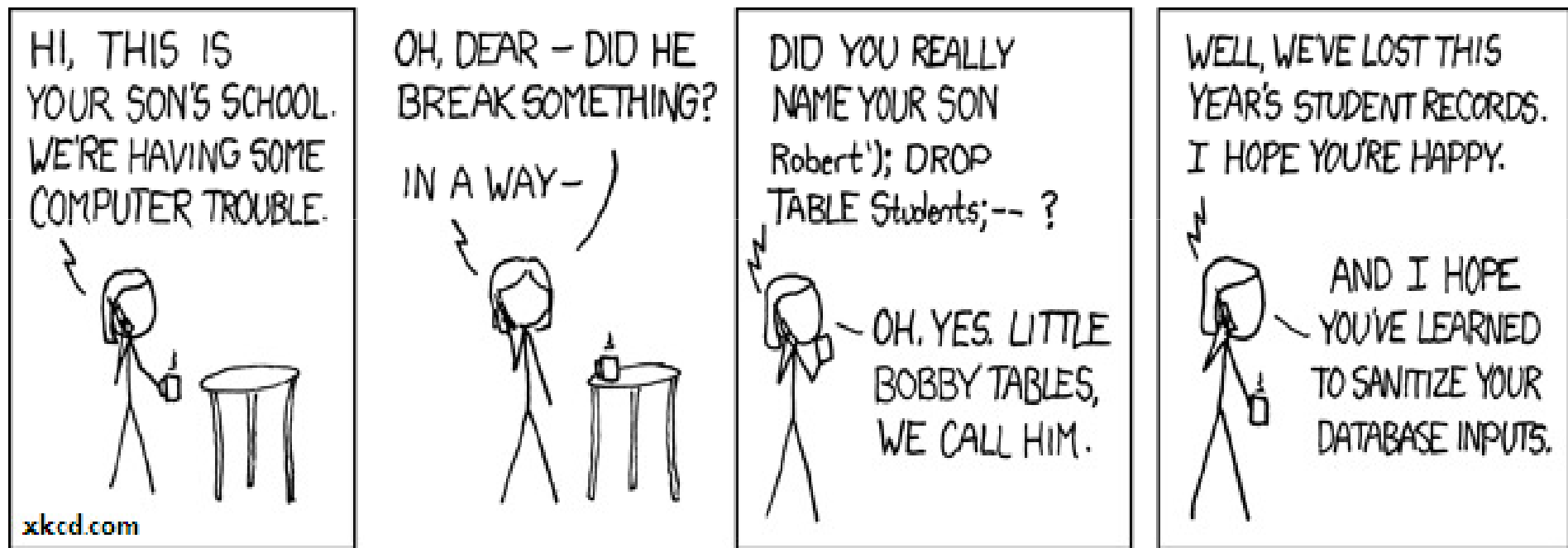
Prevention Mechanisms

- Data validation
- Stored Procedures
- Parameterized queries
- Command / Parameter objects
 - ▶ Strongly typed API
- Least Privilege

Data Validation

- Best to limit input to specific format
 - ▶ E.g. 9 digits for Id
 - ▶ Email address
 - ▶ Etc.
- Can use Regular Expressions
- But not always possible
 - ▶ Sometimes need to accept free text
 - ▶ E.g. comments, forums, etc

Parent Injection – Exploits of a Mom



Data Validation

- Ensure parameter types
 - ▶ E.g. numeric fields must be numeric
- Size
- Range
 - ▶ E.g. $0 < \text{age} < 120$
- Escape special characters
 - ▶ E.g. Quotes
- Block SQL keywords
 - ▶ E.g. UNION SELECT, INSERT etc.

Data Validation

- Best Practice: Whitelist allowed patterns
- Don't Blacklist blocked patterns/characters
 - ▶ Never complete
 - ▶ Hard to maintain
 - ▶ May affect performance...
- Blacklist not best – but can block attacks
 - ▶ Assuming specific attack was defined
- BUT.... Does it work??



OWASP Classic Smuggling

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

The Beerbelly...



General Smuggling Attacks

- Based on sneaking data where prohibited
- Smuggling avoids detection or prevention
 - ▶ Even against mechanisms that look for it
- Bad data looks good
- Malicious data does not yet exist
 - ▶ At least not in context of validation
- Cannot be detected with standard checks
 - ▶ By definition

HTTP Request Smuggling

- Discovered by Amit Klein et al. in 2005
- Based on discrepancies in parsing HTTP
- Differences in handling malformed requests
- Attacker can bypass protection mechanisms
- Causes devices to "see" different requests
- Usually not detected by IDS/IPS, WAF ...



OWASP Introducing SQL Smuggling

http://www.ComsecGlobal.com/Research/SQL_Smuggling.pdf

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Definition

- SQL Injection that evades detection
 - ▶ Even when searched for
- Exploits differences of interpretation
- Attack does not exist in validation context
- Accepted by DB server as valid

Characteristics

- Malicious strings not present
- Cannot be found by validation
- WAF and IDS/IPS mostly do not help
- Application checks do not work
- Evades Blacklists
- May be mitigated by architecture / design



"GOT ONE!"



OWASP

Common SQL Smuggling

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Platform-Specific Syntax

- Non-standard extensions to ANSI SQL
- Might not be recognized by validations
- E.g. MySQL backslash ("\"") escaping
 - ▶ Simply doubling quotes doesn't work:
 - ▶ "\"' translates to "\"'"
 - ▶ MySQL sees: "\"'"
- E.g. Who blocks [MS-SQL] OPENROWSET?

Signature Evasion

- Many validations search for known strings
 - ▶ E.g. INSERT, DELETE, UNION SELECT, etc.
- Numerous ways to evade patterns
 - ▶ Innovative use of whitespace
 - ▶ Inline comments (using /*...*/)
 - ▶ Different encodings
 - ▶ Dynamic concatenation/execution of strings
 - E.g. CHAR() or "EXEC ('INS' + 'ERT INTO...')"



OWASP

Unicode Smuggling

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Homoglyphs

- Many Unicode characters “look like” others
- E.g. Ā (U+0100) is similar to A (U+0041)
 - ▶ Stronger homoglyphs look identical
- Visually misleading
 - ▶ Can be dependant on font
- Usually mentioned as user-misdirection
- Referred to in context of IDNs

00A0	NBSP	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY	®	¯
00B0	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
00C0	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
00D0	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
00E0	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
00F0	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ
U+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0100	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Ĉ	ĉ	Ċ	ċ	Č	č	Ď	ď
0110	Đ	đ	Ē	ē	Ė	ė	É	é	Ę	ę	Ě	ě	Ĝ	ĝ	Ğ	ğ
0120	Ġ	ġ	Ģ	ģ	Ĥ	ĥ	Ħ	ħ	Ĩ	ĩ	Ĭ	ĭ	Ĵ	ĵ	Ĺ	ĺ
0130	Į	į	Ĳ	ĳ	Ĵ	ĵ	Ķ	ķ	κ	Ĺ	ĺ	Ł	ł	Ł	ł	Ł
0140	Ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł	ł
0150	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ	Œ	œ
0160	Š	š	Š	š	Š	š	Š	š	Š	š	Š	š	Š	š	Š	š
0170	Ů	ů	Ů	ů	Ů	ů	Ů	ů	Ů	ů	Ů	ů	Ů	ů	Ů	ů
0180	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ	Ɓ
0190	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
01A0	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
01B0	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
01C0	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
01D0	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ
01E0	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ	Ɔ

Character Set Support

- Servers can support translation from Unicode to Localized character sets
- Local charsets do not contain all Unicode
 - ▶ E.g. Ā not in Windows-1255
 - ▶ E.g. Ɑ (U+05D0) not in latin1
- So what happens?

Homoglyphic Transformation

- If a character is “forced” to local charset:
 - ▶ Error
 - ▶ Character is dropped
 - ▶ Automatic translation
- Translation occurs if similar character exists
- Based on “best fit” heuristic
- E.g. \bar{A} is forced to A

But \bar{A} is not A !

[illegible]

Exploit Scenario

- Attacker sends U+02BC
- Application/WAF search for quote U+0027
- Does not exist!
- Database “forces” input to local charset
- U+02BC → quote... on the database!
- Now there’s quote, get some SQL Injection!

Analysis

- Characters created by DB
- Quote does NOT exist before
- Can bypass filters and get a quote to DB
- Same with many other characters
- Can't find a quote if it's not there
- Validation CANNOT work!



OWASP Applicability

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

So, How Common IS This?

- Well, not very...
- BUT it does exist
- Originally discovered at client

Unicode-based Smuggling

■ Depends on:

- ▶ Dynamic SQL concatenation (can be in SP)
- ▶ Validation based on Blacklists
- ▶ Unicode forced into local charset
- ▶ DB support of homoglyphic transformation...
 - So far:
 - MS-SQL
 - MySQL Connect/J (old version)

On The Other Hand...

- SQL Smuggling is more common
- Aspects exist in most systems
- It is likely there are other issues to be discovered
- Most blacklists can be penetrated



OWASP

Recommendations & Conclusion

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document
under the terms of the OWASP License.

The OWASP Foundation
<http://www.owasp.org>

Recommendations

- Context-based validation
 - ▶ Relate to DB attributes
- White-list known characters
- Avoid any dynamic SQL
- Do not translate character sets
- See http://www.ComsecGlobal.com/Research/SQL_Smuggling.pdf for more information

Conclusion

- Input validation is not always enough
- SQL Smuggling can get through
- Blacklists don't work
 - ▶ Besides being inefficient
- Best Practices are there for a reason!
- Time to look at the DB platform a little more closely...

Thank you!

[http://www.ComsecGlobal.com/
Research/SQL_Smuggling.pdf](http://www.ComsecGlobal.com/Research/SQL_Smuggling.pdf)

Questions?

douglen@hotmail.com