

[NEWS] Oracle Forms SQL Injection

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-04/0074.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 04/13/05

To: list@securiteam.com

Date: 13 Apr 2005 13:53:15 +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 Forms SQL Injection

SUMMARY

<<http://www.oracle.com/technology/products/forms/index.html>> Oracle Forms 10g is "Oracle's award winning Web Rapid Application Development tool, part of the Oracle Developer Suite 10g".

All Oracle Forms applications are by default vulnerable to SQL Injection.

DETAILS

Vulnerable Systems:

- * Oracle Forms versions 3.0 up to 10g (C/S and Web)

Immune Systems:

- * Oracle Applications version 11.5.9 or newer

There is an (ancient often forgotten) Oracle Forms feature called "Query/Where" that allows users to modify existing SQL statements. This feature is quite a useful feature for power users but also dangerous due to the fact that every forms user can use it to execute arbitrary SQL statements.

Short demonstration of Oracle Forms SQL Injection:

Securiteam: [NEWS] Oracle Forms SQL Injection

1. Start a Forms module and switch to the query mode
2. Enter a colon (:) or ampersand (&)
3. An empty Query/Where windows pops up
4. Enter an SQL statement

The following statement sends the result of the SQL statement: select username from all_users where rownum=1 to a foreign (or internal) web server.

The web server of the attacker now contains the result of the custom query:
192.168.2.200 -- [14/Feb/2005:10:42:20 +0100] "GET /SYS HTTP/1.1" 404 209

Impact:

The impact of the SQL injection depends on the architecture of your Forms application.

If the Oracle user used by the Forms application has DBA privileges (like Oracle Applications), EVERY user can select any data in the database. Never set the value FORMSxx_RESTRICT_ENTER_QUERY to FALSE in an existing Oracle Applications environment because every user can execute ANY statement and see ANY data.

If your Forms application implements an own user concept (e.g. own user table including passwords) it is possible that, other users could see the data (e.g. their accounts/passwords/).

In all other cases the Forms user can still execute PLSQL packages granted to public like utl_http.ata.

Workaround:

Set the undocumented environment variable FORMSxx_RESTRICT_ENTER_QUERY=true (FORMS60_RESTRICT_ENTER_QUERY for Forms 6.x, FORMS90_RESTRICT_ENTER_QUERY for Forms 9.x/10g) and restart the Forms server. This environment variable disables the possibility of using the query/where functionality.
or only if really need Query/Where:

Write a PRE_QUERY/ON-ERROR-trigger for EVERY input field and validate the entire input for EVERY Oracle Forms module (*.fmb)

Disclosure Timeline:

- * 7-oct-2003 Oracle secalert informed
- * 7-oct-2003 Bug confirmed

ADDITIONAL INFORMATION

The information has been provided by <mailto:alexander.kornbrust@web.de> Alexander Kornbrust.

The original article can be found at:

<http://www.red-database-security.com/wp/sql_injection_forms_us.pdf>

Securiteam: [NEWS] Oracle Forms SQL Injection

http://www.red-database-security.com/wp/sql_injection_forms_us.pdf

=====

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@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

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.