APEX-Security – Ein Überblick für Einsteiger

Raphael Salguero
ORDIX AG
Paderborn

Schlüsselworte
APEX Anwendungsentwicklung Security Autorisierung Authentifizierung SQL-Injection Cross-Site-Scripting

Einleitung
Oracle bietet mit Oracle Application Express (APEX) eine kostenlose Möglichkeit zur Erstellung von Web-Applikationen an. Diese Applikationen sind mit Oracle APEX schnell entwickelt, allerdings kommen wichtige Sicherheitsaspekte meist zu kurz. Dass hierbei oft nachlässig mit wichtigen Unternehmensdaten umgegangen wird, ist den Entwicklern oftmals gar nicht bewusst. So reicht einem potentiellen Angreifer bereits die Ergänzung eines Eingabefeldes um wenige Wörter aus, um mittels SQL-Injection an unerwünschte Datensätze zu gelangen. Durch das Aufzeigen diverser Sicherheitslücken in der Theorie und in der Praxis wird der APEX-Entwickler sensibilisiert.

Im Fokus der Folien stehen hierbei grundlegende Informationen zu Autorisierungs- und Authentifizierungsmöglichkeiten sowie SQL-Injection und Cross-Site-Scripting. Auch das Thema Verschlüsselung im Zusammenhang mit dem PL/SQL Embedded Gateway (EPG) oder dem Oracle Rest Data Service (ORDS) spielt beim Thema APEX-Security eine Rolle.
Anhand von Beispielanwendungen werden diese Grundlagen praxisnah demonstriert. Dabei werden gleichzeitig Hinweise geliefert, wie die Sicherheit der Applikationen bereits mit einfachen Mitteln erhöht werden kann.

Authentifizierungsmöglichkeiten
APEX bietet mehrere Möglichkeiten der Authentifizierung eines Anwenders an der Applikation an. Dabei reichen die Varianten von einer „Open Door Credentials“-Authentifizierung bis zu einer selbst konfigurierten „Single Sign-On“-Variante.
Neben dem seit APEX 18.1 verfügbaren „Social Login“ bietet APEX auch die Verwendung von internen Applikations-Benutzern oder einem LDAP-Server an. Bei der Verwendung eines LDAP-Servers ist beispielsweise zu beachten, dass der APEX eigene Button „SSL verwenden“ nicht wie gewünscht funktioniert und die SSL-Verschlüsselung in Kombination mit einem LDAP-Server manuelle Schritte erfordert.
In diesem Vortrag werden verschiedenen Möglichkeiten der Authentifizierung mit ihren möglichen Vor- und Nachteilen demonstriert.

Autorisierungsmöglichkeiten
Die Verwendung von Autorisierungs-Schemata ist die einfachste Möglichkeit, um innerhalb einer Applikation vor einem Zugriff auf sensible Daten zu schützen. Hierbei lassen sich sowohl einzelne Datensätze und Items, als auch ganze Regionen, Seiten oder Menüpunkte mit einem Autorisierungs-Schema vor unbefugten Zugriffen schützen. Die Autorisierungs-Schemata können beispielsweise auf einer SQL-Abfrage, einer Gruppenzuordnung oder auch auf den Wert eines Page-Items basieren. Auch hier werden verschiedene Möglichkeiten sowie die Vor- und Nachteile im Vortrag demonstriert.

Neben diesen zugriffsschützenden Maßnahmen gibt es einige relevante Sicherheitslücken, die durch Entwicklungsfehler begünstigt werden. Hierzu gehören beispielsweise Folgende Sicherheitslücken:

- SQL-Injection

- Cross-Site-Scripting
- Cross-Site-Request-Forgery
- Insecure Direct Object References

SQL-Injection
SQL-Injection ist eine seit Jahren bekannte Sicherheitslücke von Web-Applikationen und wird durch die falsche Verwendung von Variablen in einem SQL-Statement begünstigt. Bei dieser Sicherheitslücke wird ein vorbereitetes SQL-Statement durch eine Benutzereingabe verändert. So können zum Beispiel unerwünschte Datensätze ausgelesen oder gar unerwünschte Datenbankkommandos ausgeführt werden.
Das Risiko einer SQL-Injection existiert vor allem bei der Verwendung von dynamischen SQL-Statements, die aus diversen Gründen keine Verwendung von sicheren Bind-Variablen unterstützen.
Das folgende Beispiel zeigt exemplarisch die unerwünschte Erweiterung eines SQL-Statements, welches zur Folge hat, dass alle Mitarbeiterinformationen ausgegeben werden:

```
SELECT *
  FROM MITARBEITER
 WHERE MITARBEITER_NR = ' ' OR 1 = 1 – ' AND ABTEILUNGS_NR != 1;
```

Cross-Site-Scripting
Das dynamische Erzeugen von Inhalten einer Website erhöht auch das Risiko einer Cross-Site-Scripting-Attacke. Beim Cross-Site-Scripting wird Schadcode (z.B. in Form von JavaScript) im Browser des Benutzers ausgeführt. Mit einem solchen Angriff könnten unter anderem die Cookies des Benutzers, also auch weitere Session-Informationen abgegriffen werden.

Cross-Site-Request-Forgery
Bei dem Cross-Site-Request-Forgery handelt es sich ebenfalls um einen Angriff auf den Benutzer der Applikation. Hierbei wird der Benutzer dazu gebracht, einen manipulierten HTTP-Request abzusenden, welcher ebenfalls Session-Informationen aus den Cookies des Benutzers abzugreifen versucht.
Mit diesen Session-Informationen wird der manipulierte HTTP-Request autorisiert. Ein typisches Beispiel für einen Cross-Site-Request-Forgery sind unbewusst veröffentlichte Beiträge in sozialen Netzwerken. Dass solch ein Angriff auch zu unerwünschten Handlungen in einer APEX-Applikation führen kann, wird im Vortrag demonstriert.

Insecure Direct Object References
Unsichere direkte Objektreferenzen stellen speziell bei APEX-Applikation eine Gefahr dar und sollten insbesondere bei der Verwendung von Master-Detail-Reports berücksichtigt werden. Gefährlich ist hierbei die Übergabe von Primary Keys zur Identifizierung eines Datensatzes über die URL. Standardmäßig kann diese Übergabe ohne großen Aufwand von einem Benutzer verändert und so der Zugriff auf Datensätze ohne Berechtigung erfolgen.

Verschlüsselung
Oracle APEX kann auf unterschiedliche Art und Weise betrieben werden. Die einfachste Methode stellt hier die Verwendung des PL/SQL Embedded Gateways (EPG) dar. Hierbei wird der Oracle Datenbank-Listener auch als Web-Listener verwendet. Standardmäßig wird beim EPG kein HTTPS, also keine SSL-Verschlüsselung verwendet. Um SSL verwenden zu können, ist die Oracle Enterprise Edition mit der Advanced Security Option notwendig.
Eine lizenzfreie Alternative hierzu bietet die Verwendung des Oracle Rest Data Services (ORDS) in Verbindung mit einem Webserver (z.B. der Apache Tomcat). Bei dieser Architektur bleibt die Kommunikation zwischen der APEX-Applikation und der Datenbank zwar weiterhin unverschlüsselt,

dafür lässt sich aber mit einfachen Mitteln die Kommunikation zwischen dem Webserver und dem Browser des Benutzers verschlüsseln.

Zusammenfassung
Oracle APEX bietet bereits standardmäßig viele Möglichkeiten, die Web-Anwendung sicher zu gestalten. Allerdings ist für eine wirklich sichere APEX-Entwicklung der Entwickler selbst gefragt.
Der Vortrag soll an dieser Stelle lediglich einen Überblick für Einsteiger liefern und die Gefahr, welche von den bekannten Sicherheitslücken ausgehen kann, deutlich machen.

Kontaktadresse:
Raphael Salguero
ORDIX AG
Karl-Schurz Straße 19a
D-33100 Paderborn

Telefon: +49 (0) 5251 1063-0
Fax: +49 (0) 180 1673490
E-Mail info@ordix.de
Internet: www.ordix.de