FMW 11g End To End Security

Olaf Heimburger
Oracle Deutschland B.V. & Co. KG
Berlin

Schlüsselworte:

Fusion Middleware, Security, Datenbank

Einleitung

Heutige Anforderungen an Datensicherheit werden immer komplexer und wichtiger. Diese Funktionalität darf aber nicht nur den Entwicklern überantwortet werden, sondern muss auch im produktiven Einsatz den aktuellen Bedürfnissen angepasst werden können. In diesem Vortrag wird anhand einer vollständigen Anwendung gezeigt, wie diese Anforderungen in allen Schichten einer Anwendung implementiert werden können und während der Laufzeit, also ohne Neuprogrammierung, geändert werden können.

Moderne Architekturen

Unsere heutigen, modernen Anwendungsarchitekturen bestehen aus mindestens 2 Schichten und werden durch die Verwendung von weiteren Schichten zunehmend komplexer.

Standards für die Sicherheit

Durch die unterschiedlichen Anwendungsgebiete haben sich für die Datensicherheit viele Standards etabliert. Der bekannteste, weil verbreitetste, ist sicherlich die Secure Socket Layer (SSL) oder Transport Level Security (TLS). Je Anwendungsschicht kommen andere Standards wie ... hinzu. Alle diese Standards haben gemein, dass keiner alleine stehen sollte und in der Kombination zu einer runden Lösung führen.

Eine sichere Architektur sieht SSL als Absicherung der Kommunikationsleitungen vor und verwendet auf den weiteren Ebenen die passenden Standards zur Autorisierung/Authentifizierung aber auch zur Verschlüsselung. Passend kann hier auch sein, dass die teilnehmenden Anwendungsteile noch nicht so weit sind oder sogar höhere Anforderungen haben. Zu den passenden Standards gehören AES-Verschlüsselung, WS-Security, WS-Encryption, SAML und so weiter.

Zukünftige Anforderungen

Gerade bei der Verschlüsselung ist eine Lösung solange sicher bis sie in vertretbaren Aufwand geknackt werden kann. Dieser Zeitraum ist durch steigende Rechenleistung oder neue Wege in der Ressourcenbeschaffung (durch Cloud Computing oder verteilte Rechnernetze wie bei Seti@Home) schwer vorherzusagen und im schlimmsten Fall schon bei der ersten Veröffentlichung der Anwendung nicht mehr sicher. Noch schlimmer sind gesetzliche Anforderungen oder andere Regelwerke (zum Beispiel das Bundesdatenschutzgesetz in der Fassung von 2009 oder die ISO 27001). Durch die Globalisierung kommen auf eine beliebige Anwendung urplötzlich Anforderungen hinzu, an die man vorher gar nicht denken konnte.

Eine Anwendung

Der Einfachheit halber gehen wir von einer hypothetischen Anwendung aus. Diese Anwendung hat folgende Komponenten:

- eine grafischen Oberfläche
- einen Web Service
- eine Session Facade
- eine Schnittstelle zu einer Datenbank
- eine Datenbank

Für diese Anwendung gilt folgende Anforderung: Auf jeder Ebene sollte nur die Identität des Anwenders verwendet werden. Technische Anwender sind nicht oder nur mit eingeschränkten Rechten zu verwenden.

Datenbankschicht

Da keine technischen Anwender erlaubt sind, müssen wir erreichen, dass der Anwender auch in der Datenbank bekannt ist und dort verwendet wird. Üblicherweise wird zwischen der Datenbank und einem Application Server eine Datenquelle (DataSource) konfiguriert. Für diese DataSource braucht man einen Datenbankbenutzer. Dieser ist aber nicht der aktuelle Anwender und schon bekannt bevor die Anwendung an die eigentlichen Anwender ausgeliefert wird. Das klingt nach einem Teufelskreis und ist es bei den meisten Datenbank auch.

Bei einer Oracle Datenbank kann dieser Teufelskreis durch die Anwendung der Funktionalität Proxy Authentication erreichen. Es wird ein sogenannter Proxy User mit sehr wenigen Rechten in der Datenbank angelegt. Alle richtigen Anwender bekommen das Recht sich über diesen Proxy User anmelden zu dürfen zugewiesen. Mit diesem Mechanismus werden gleich mehrere Probleme gelöst. Der Proxy User hat nur eingeschränkte Rechte und kann auf keinerlei Daten zugreifen. Der Proxy User kann auch im Application Server für die Konfiguration der DataSource verwendet werden. Und schließlich kann sich der richtige Anwender in der Datenbank anmelden und mit den ihm zugeordneten Rechten auf die richtigen Daten zugreifen. Dabei ist es egal welche Datenbankverbindung (ODBC, JDBC oder SQL*Net) gewählt wird.

Die Schnittstelle zur Datenbank

Zur Realisierung der Schnittstelle zur Datenbank (Mapping Layer) gibt es mehrere Lösung auf dem Markt. Im Umfeld der Oracle Fusion Middleware 11g sind die Bibliotheken ADF Business Components oder EJB 3.0/Java Persistence API (JPA) sehr gut geeignet. Die Schnittstelle zur Datenbank im Application Server verwendet die DataSource mit den oben angegebenen Werten (also Proxy User und Passwort).

Durch den Proxy User ist der Zugriff auf alle Daten solange ausgeschlossen, bis der richtige Anwender mitgeliefert wird. Die Verwendung der DataSource ist gewährleistet, benötigt aber die zusätzliche Information des realen Anwenders (z.B. Name und Passwort) bevor auf die richtigen Daten zugegriffen wird. Diese Information muss der verwendeten Bibliothek mitgegeben werden.

Die Session Facade

Die Session Facade wird häufig verwendet um den Zugriff auf die unterliegende Implementierung der Mapping Layer zu verbergen. Diese muss auch dafür sorgen, dass die Information des Anwenders an die Mapping Layer weitergereicht wird.

Der Web Service

Web Service Schnittstellen helfen die Funktionen einer Anwendung über einen plattform-unabhängigen Mechanismus einem breiteren Spektrum von Anwendungen zur Verfügung gestellt werden. Durch die Verwendung von XML als Transporttechnologie ist die Vertraulichkeit der Daten zunächst nicht gegeben. Um dies zu erreichen wurden ergänzende Standards wie WS-Security, WS-Encryption, etc. definiert. Je nach Anforderung kann ein WebService einen oder mehrere dieser Standards implementieren.

Die grafische Oberfläche

Die grafische Oberfläche ist sehr häufig der einzige Weg den der Anwender bedient um Daten erfragen, modifizieren und speichern zu können. In unserem Beispiel meldet sich der Anwender an dieser Stelle an und kann über weitere Masken die Daten aus der Datenbank erfragen, modifizieren und speichern. Er meldet sich aber nur einmal an. Als besondere Herausforderung muss der WebService in die Oberfläche eingebunden werden.

Die Oberfläche wird mit ADF Faces implementiert. Für die Zugriffkontrolle wird ADF Security verwendet. Der Aufruf des WebService wird über ein ADF DataControl aufgerufen. Durch das DataControl wird die Implementierung an ADF delegiert und es ist lediglich die Konfiguration des Zugriffs aber auch der Security erforderlich.

Die Identität des Anwenders

Der Anwender meldet sich über die grafische Oberfläche an. Es wird in der Anwendung eine Identität erstellt. Diese Identität muss nun an das WebService DataControl weitergereicht werden. Je nach Einstellung muss der geforderte Authentifizierungsmechanismus (wie WS-Security UsernameToken, X.509 Zertifikat oder SAML Token) des WebService verwendet werden.

Kann der WebService erfolgreich identifiziert werden, steht die Identität nahezu sofort der darunterliegenden Session Facade und der Mapping Layer zur Verfügung. Die Mapping Layer kann diese Identität vor jedem Datenbankzugriff als realen Anwender an die Datenbank übergeben und damit Zugriff auf die Daten.

Die Implementierung der Identität

Jede Technologie hat eigene Möglichkeiten eine Identität zu implementieren. In Java wird dies durch den *Java Authentication and Authorization Service* (JAAS) erreicht. Die Identität wird als *Principal* in einem *Subject* erreicht. Beide werden bei einer Anmeldung durch ein *LoginModule* erstellt und stehen während der gesamten Transaktion zur Verfügung.

In unserem Beispiel wird diese durch die Anmeldung in der ADF Faces Anwendung mit ADF Security erstellt und über das DataControl an den WebService weitergegeben. Der WebService erstellt eine ähnliche Kombination von Subject und Principal welche damit der Session Facade sowie der Mapping Layer zur Verfügung steht. Die Mapping Layer kann diese nun an die Datenbank weitergeben.

So weit, so gut. Was fehlt ist im Minimalfall, also bei der Verwendung der einfachen Username/Passwort-Kombination, das Passwort. Aber genau das Passwort wird von JAAS nicht zur Verfügung gestellt.

Dieses Manko wird durch die Oracle Platform Security Services (OPSS) behoben im WebLogic Server 10.3 implementiert. OPSS ist flexibel und modular aufgebaut und erlaubt die Integration beliebiger LoginModule, die helfen die notwendigen Subjects und Principals zu erstellen. In Kombination mit dem Oracle Web Service Manager erlaubt OPSS die Identitäten zwischen verschiedenen Schichten in der Anwendung, aber auch an externe Anwendungen weiterzuleiten, ohne dass der Anwender sich erneut anmelden muss.

Die Beispielanwendung

Die vorgestellte Beispielanwendung zeigt wie die einzelnen Schichten implementiert und dabei die Identität erzeugt und weitergeleitet. Jede Schicht wird dabei einzeln betrachtet und gezeigt das diese Identität notwendig ist um die Daten zu gelangen. Besonders hervorzuheben ist hier, dass diese wirklich für jede Schicht gilt und nicht umgangen werden kann.

Kontaktadresse:

Olaf Heimburger
Oracle Deutschland B.V. & Co. KG
Schloßstr. 2
D-13507 Berlin

Telefon:	+49 (0) 30 435 795-160
Fax:	+49 (0) 30 435 795-419
E-Mail	olaf.heimburger@oracle.com
Internet:	blogs.oracle.com/olaf