Oracle Forms Single Sign On

Authenticatie voor Oracle AS-omgevingen

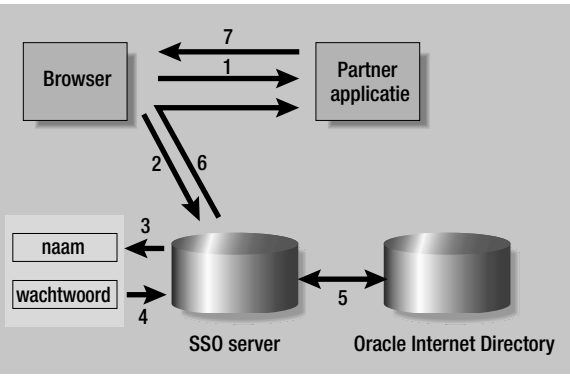
Oracle's Single Sign-On (SSO) is geïntroduceerd in Oracle Application server release 1 als onderdeel van Oracle Portal. In de Oracle Application server release 2 werd SSO losgetrokken van Oracle Portal en werd SSO een onderdeel van de 'infrastructure' component waardoor Oracle Portal niet per se meer benodigd was. Tevens ging SSO vanaf release 2 gebruik maken van Oracle's Internet Directory (OID).

Oracle SSO wordt gebruikt als centraal authenticatiesysteem voor veel applicatieomgevingen binnen de Oracle Application Server. De gebruikersgegevens zelf zijn geregistreerd in Oracle Internet Directory. Door Oracle SSO is het dus mogelijk om de gebruikeradministratie centraal uit te voeren waardoor veel (beheer)kosten bespaard kunnen worden. Oracle SSO wordt gebruikt in diverse Oracle tools, zoals Oracle Portal, Oracle Forms, Oracle Reports, Oracle Discoverer maar ook in Java/J2EE applicaties die binnen de Oracle Application Server draaien. Let er wel op dat Oracle SSO onderdeel is van de (dure) Enterprise Edition van Oracle Application Server. Meestal vindt in Oracle SSO de gebruikers-authenticatie plaats met behulp van een gebruikersnaam en wachtwoord, maar deze kan ook plaatsvinden via SSL met een client en server X.509 certificaat uitwisseling waardoor er geen gebruikersnaam en wachtwoord behoeft te worden opgegeven. Dit mechanisme wordt Public Key Infrastructure (PKI) genoemd.

Zoals we in ons vorige artikel over Oracle Internet Directory uiteengezet hebben (zie de vorige editie van Optimize), betekent het inzetten van OID - en dus SSO - dat gebruikersnamen en wachtwoorden niet meer in de database worden vastgelegd. In OID worden gebruikers en groepen centraal geregistreerd om vervolgens gedeeld te worden tussen meerdere applicaties, bijvoorbeeld Oracle Portal, Forms en Collaboration Suite. Applicaties die gebruik maken van Oracle SSO worden partner applicaties genoemd en behoeven zelf geen gebruikersnamen en wachtwoorden vast te houden. Oracle SSO wordt ook gebruikt voor applicaties die wel een eigen gebruikersnaam/wachtwoord mechanisme hebben. Deze applicaties worden externe applicaties genoemd. Oracle SSO houdt voor externe applicaties een lijst bij met gebruikersnaam/wachtwoord combinaties die gebruikt kunnen worden om in te loggen op deze applicaties.

Werking van SSO

Oracle SSO gebruikt cookies om informatie over de identiteit van de gebruiker vast te leggen op de browser client. Oracle SSO werkt dan ook niet als de gebruiker cookies niet wil accepteren. Wanneer een gebruiker een SSO-enabled applicatie voor de eerste keer benadert, wordt hij of zij doorgesluisd naar Oracle SSO. Deze controleert of de gebruiker een geldige SSO cookie heeft. Indien dit niet het geval is krijgt de gebruiker



1 Gebruiker start Partner App, App bepaalt dat gebruiker nog niet gevalideerd is (geen applicatie cookie)
2 App sluist gebruiker door naar SSO server
3 SSO server opent naam/wachtwoord pagina
4 SSO ontvangt naam en wachtwoord
5 SSO valideert naam en wachtwoord in OID
6 SSO zet een SSO cookie en sluist de gebruiker terug naar App met versleutelde token
7 App zet applicatie cookie

Figuur 1. Het mechanisme van OracleAS Single Sign-On

een SSO login scherm met het verzoek om een gebruikersnaam en wachtwoord in te geven. Na een succesvolle validatie van de gebruikersgegevens wordt alsnog een SSO cookie in de browser gezet. De cookie is versleuteld door Oracle SSO, waardoor het niet gezet of gelezen kan worden door een derde partij. Tevens verlopen de SSO cookies na een bepaalde periode of wanneer de gebruiker de browser afsluit. In onderstaand schema wordt het één en ander schematisch weergegeven.

Merk op dat alle interactie tussen Oracle SSO, de browser clients, en de applicaties via standaard HTTP verlopen. Er zijn dus geen speciale eisen voor de client browser anders dan dat cookies geaccepteerd moet worden. Het is tevens raadzaam

Oracle SSO gebruikt cookies om informatie over de identiteit van de gebruiker vast te leggen op de browser client

om SSL te gebruiken in het verkeer tussen Oracle SSO en de client browser om te voorkomen dat derden gebruikersnamen en wachtwoorden kunnen onderscheppen.

Nadat een gebruiker gevalideerd is en in een SSO cookie gezet is, wordt de gebruiker door Oracle SSO teruggesluisd naar de applicatie waarbij een versleutelde 'token', met daarin de identiteit van de gebruiker, in de URL wordt gezet. Deze 'token' is versleuteld met een sleutel die alleen gedeeld wordt met de betreffende applicatie. Wanneer de applicatie de URL ontvangt en ontcijfert, kan het bepalen of de gevalideerde gebruiker toegang verleend moet worden tot de applicatie. Indien de gebruiker toegang krijgt tot de applicatie wordt een applicatie cookie gezet in de client browser waardoor de gebruiker vervolgens door de applicatie geïdentificeerd kan worden zonder dat deze steeds terug behoeft te gaan naar Oracle SSO.

Oracle Forms

Laten we nu kijken naar de concrete toepassing van Oracle SSO in Oracle Forms applicaties. We hebben het dan over webforms applicaties, die gebruik maken van de OracleAS Forms services component van de Oracle Application Server, en niet over de client/server forms. De SSO integratie met Forms is tot stand gekomen bij de introductie van de Forms services servlet, en is al vanaf Forms 6i (patch 5) toepasbaar. Voor het toepassen van SSO in Oracle Forms heeft Oracle een tweetal extra componenten geïntroduceerd:

- Mod_osso - dit is een extensie van de Oracle HTTP server die zich als een SSO partner applicatie gedraagt. Applicaties die gebruik maken van mod_osso, zoals servlets, kunnen op deze wijze gebruikers valideren met behulp van het Oracle SSO framework zonder dat ze hiervoor specifieke logica behoeven te bevatten. OracleAS Forms Services en de OracleAS Reports Services gebruiken mod_osso om zich als partner applicatie in de SSO server te registreren.

- Resource Access Descriptor - Een OracleAS Forms services applicatie heeft een connectie nodig met de database. Deze database connect informatie worden per SSO gebruiker vastgelegd in OID met behulp van een zogenaamde Resource Access Descriptor (RAD). Het is overigens ook mogelijk om gebruik te maken van een default RAD die door meerdere Forms gebruikers wordt gebruikt.

Interactie Oracle Forms en SSO

Met het bovenstaande in ons hoofd kijken we nu in detail naar de interactie tussen een Oracle Forms applicatie en de SSO server. Een Oracle Forms applicatie, of beter gezegd een Forms services applicatie, wordt benaderd met een URL zoals bijv. http://server.whitehorses.nl:7777/forms90/f90servlet?config=mijnapp. In deze URL wordt de betreffende Forms applicatie ('mijnapp') gespecificeerd met de config parameter, die refereert aan een 'Named Configuration' in de Forms Services configuratie file (formsweb.cfg).

De Forms servlet leest de SSO parameters (zoals het adres van OID) uit deze configuratie file voor de betreffende applicatie en sluist de gebruiker door naar de mod_osso module. De mod_osso module controleert vervolgens of de gebruiker voor deze applicatie URL al gevalideerd is. Indien dit niet het geval is wordt de gebruiker doorgesluisd naar de SSO server. De SSO server controleert of er een SSO cookie aanwezig is voor deze browser sessie. Indien dit niet het geval is krijgt de gebruiker de SSO login pagina. Nadat de gebruiker succesvol

Figuur 2. De login pagina van de OracleAS Single Sign-On server

gevalideerd is wordt de gebruiker weer teruggeleid naar de Forms servlet.

De Forms servlet leest vervolgens de Resource Access Desciptor uit OID en genereert de HTML startpagina voor de Forms Applet. De database connect informatie wordt in de servlet sessie bewaard op de applicatie-tier en wordt niet vrijgegeven aan de browser. Vervolgens maakt de Forms Applet contact met de Forms Listener Servlet voor alle verdere HTTP of HTTPS communicatie.

Na een succesvolle validatie van de gebruikersgegevens wordt alsnog een SSO cookie in de browser gezet

Omgaan met accounts

Het toepassen van Oracle SSO in bestaande Forms applicaties vereist enige aanpassingen aan deze applicaties. Door het verplaatsen van de gebruikersadministratie naar Oracle Internet Directory wordt er immers nog maar één (of hooguit enkele) database accounts gebruikt in plaats van individuele Oracle accounts. Een bijkomend effect is dat hierdoor alle applicatie gebruikers dezelfde database rollen gaan gebruiken, waardoor de gehele beveiliging van de applicatie anders moet worden opgezet.

We moeten dus een geheel nieuwe weg inslaan waarbij we ons concentreren op de SSO account in plaats van de database account. Zoals we in ons vorige artikel over Oracle Internet Directory schreven zijn een drietal scenario's mogelijk bij het inzetten van Oracle Internet Directory (OID):

1. Geheel overstappen op OID waarbij alle bestaande database accounts en rollen over boord worden gezet, of
2. Gedeeltelijk overstappen, waarbij alleen de database accounts verdwijnen maar de database rollen behouden blijven, of
3. Alle database accounts en rollen behouden, naast een volledige administratie in OID.

Omdat optie 3 geen optimaal scenario is en voorbij gaat aan de voordelen van OID, namelijk een centrale gebruikersadministratie, concentreren we ons in de rest van dit artikel op de scenario's 1 en 2.

Functionaliteit inbouwen

Om nu in Oracle Forms overweg te kunnen met SSO dient er één en ander in de applicatie te worden ingebouwd. Het gebruikte SSO account dient allereerst te wordt opgehaald uit Oracle SSO en moet vervolgens gedurende de gehele Oracle Forms sessie worden vastgehouden. Alle gebruikersvalidaties in de applicatie dienen vervolgens plaats te vinden met behulp van dit SSO account en niet meer met het gebruikte database account. We lopen stapsgewijs door het mechanisme heen.

1. Ophalen SSO account
De eerste stap is om te bepalen welk SSO account gebruikt is. Dit kan gelukkig eenvoudig in Oracle Forms met behulp van de Forms built-in get_application_property(sso_userid) waarmee de SSO gebruikersnaam wordt opgehaald.

2. Ophalen SSO groepen
Vervolgens moet vanuit OID de bijbehorende gebruikersgroepen worden opgehaald. Vanuit Oracle Forms wordt op een tweetal manieren met OID gecommuniceerd:
- door middel van de database packages DBMS_LDAP en DBMS_LDAP_UTIL, of
- met behulp van de te importeren Java klasse frmDirectorymanager.

Het importeren van Java classes is een nieuwe functionaliteit van Oracle Forms en biedt de mogelijkheid om Java rechtstreeks in Forms te gebruiken. Het werken met frmDirectorymanager is wat eenvoudiger dan het werken met de LDAP database packages, omdat er minder code benodigd is. In onderstaande PL/SQL code wordt verduidelijkt hoe frmDirectorymanager gebruikt wordt.

```
declare
   frmDirMan                          ORA_JAVA.JOBJECT;
   userGroupArrays              ORA_JAVA.JOBJECT;
   userGroupArray               ORA_JAVA.JARRAY;

   l_currentUser                      varchar2(2000);
   l_userGroupDN                varchar2(2000);

 begin
   – initialiseren van een frmDirMan instance
   – connectie settings voor OID kunnen in de aanroep worden gespecifi-
 ceerd
   – of staan in de config file frmDirMgr.properties
   –
   frmDirMan := frmDirectoryManager.new();
   –
   – ophalen SSO account
   –
   l_currentUser := GET_APPLICATION_PROPERTY(SSO_USERID);
   –
```

```
  – ophalen gebruiker gegevens uit Oracle Internet Directory
  –
  if not
frmDirectoryManager.loadCnAsCurrentUser(frmDirMan,l_currentUser)
  then
                                ...fout!
  end if;
  –
  – ophalen groepen voor de current SSO gebruiker
  –
  userGroupArrays :=
frmDirectoryManager.getCurrentUserGroupMemberships(frmDirMan);
  if ora_java.get_array_length(userGroupArrays) > 0
  then
    –
    – uitpluizen van de array en verwerken van de opgehaalde groepen
    –
    for i in 0 .. ora_java.get_array_length(userGroupArrays)-1
    loop
      userGroupArray :=
ora_java.get_object_array_element(userGroupArrays,i);
      if ora_java.get_array_length(userGroupArray)>0
      then
        l_userGroupDN :=
ora_java.get_string_array_element(userGroupArray,1);
        ... verdere verwerking
        ...
      end if;
    end loop;
  end if;
end;
```

3. Vastleggen van de SSO gebruiker en SSO groepen
De laatste stap is om het SSO account en de SSO groep(en) vast te houden gedurende de Forms sessie zodat deze in de diverse Forms modules gebruikt kunnen worden. Gebruik hiervoor bijvoorbeeld een eigen database package met een aantal publieke variabelen. Het voordeel van deze constructie is dat ook de database views en andere packages eventueel gebruik kunnen maken van deze informatie.
Indien nu gekozen is om nog met database rollen te werken (scenario 2) moeten deze nu geactiveerd (enabled) worden met behulp van de database package dbms_session. Hierbij kan gekeken worden naar welke gebruikersgroepen de SSO gebruiker toegewezen heeft gekregen in OID.

```
dbms_session.set_role((mijn_database_role();
```

Door gebruik te blijven maken van database rollen kan een bestaande database beveiliging intact blijven.

Valkuilen

Bij het invoeren van SSO accounts in bestaande Oracle Forms applicaties dient men op een paar valkuilen bedacht te zijn. We noemen een tweetal:

- Oracle Forms menu vormt misschien wel de grootste uitdaging bij de invoering van SSO accounts, omdat de menu autorisatie afhankelijk is van de enabled database rollen van een Oracle account. Doordat nu meerdere SSO accounts gebruik maken van hetzelfde database account met bijbehorende database rollen krijgen meerdere SSO accounts een identieke menu autorisatie. Er zijn een tweetal oplossingen voor dit probleem. Allereerst kunnen we de database rollen, die het menu gebruikt, laten bestaan in de database (zie hierboven bij stap 3). Door nu deze rollen per gebruiker selectief aan te zetten of uit te zetten op grond van de SSO gebruikersgroepen kan ook het Forms menu gebruik blijven maken van de database rollen. Een andere (ingewikkelder) oplossing is om de menu items met de Forms built-in set_menu_item_property aan- of uit te zetten. Op otn.oracle.com wordt deze optie uitgewerkt in een voorbeeld.

- Een andere uitdaging vormen de auditing kolommen Created_By en Modified_By waarin Oracle Designer automatisch het Oracle account die de rij in de tabel aanmaakt of wijzigt wegschrijft. Omdat we nu niet meer geïnteresseerd zijn in het database account moet dit veranderd worden in het SSO account. Gelukkig kent Oracle Designer de mogelijkheid het interne mechanisme te vervangen door een eigen database functie. Hiervoor moet een database functie geschreven worden die het actieve SSO account voor de sessie teruggeeft. Deze database functie kan het SSO account vinden in de hierboven genoemde database package waarin we het SSO account en de SSO groep(en) vasthouden gedurende de Forms sessie.

Conclusie

Het toepassen van SSO voor Forms applicaties is een interessante en relatief eenvoudige operatie. Naast een groot gebruikersgemak doordat de gebruiker slechts éénmaal behoeft in te loggen, werkt het centraliseren van de gebruikersadministratie kostenbesparend. De kosten van de benodigde aanpassingen aan de bestaande Forms applicaties wegen ons inziens ruimschoots op tegen de voordelen van het gebruik van Oracle Internet Directory. Deze besparing wordt verstrekt indien er ook een Oracle Portal of Oracle Collaboration Suite omgeving aanwezig is, waar immers ook al gewerkt wordt met OID. Bij nieuwbouw van Oracle Forms applicaties doet men er sowieso goed aan om te kiezen voor toepassing van SSO en OID omdat op deze wijze wordt aangesloten op Oracle's multi-tier applicatie architectuur.

Rudolf Beekman en **Mark Nikkels** zijn senior consultants bij Whitehorses (e-mail: rudolf.beekman@whitehorses.nl en mark.nikkels@whitehorses.nl).