An Oracle White Paper

July 2011

Secure Searching with Oracle Secure Enterprise Search 11g

Executive Overview

To be useful, an enterprise search engine must deal with documents from a variety of sources. Many of these documents and sources will be *protected* – available only to authorized users. We look at how Oracle Secure Enterprise Search handles such protected sources.

Introduction

Oracle Secure Enterprise Search (SES) is a product that enables you to find information within your corporate intranet by keyword or contextual searches.

To do this, it must first collect the content from diverse sources and text-index it.

The process of collecting this information is known as crawling, and is performed by a crawler.

This document looks at the various ways that crawlers can access secure content on a network, and how Oracle SES prevents access to that content by unauthorized users.

Protected Sources

Some of the information within an organization is publicly accessible. Anyone is allowed to view it, and it is therefore relatively easy for a crawler to find and index that information.

However, there are many sources that are protected: only certain users or user groups (such as "the finance department") are allowed to access that data, or parts of that data.

An easy to understand example is an email system. Within most organizations, a user's emails are considered private to that individual. While we want any user to be able to search their own emails, we do not want anybody to be able to go in and search over everybody's emails, or even any other individual's emails.

Secure crawling, secure search results

So there are two challenges. First, the crawler itself must be able to access the protected sources. It must be able to authorize itself in some manner to gain access to the sources – either by being trusted by the source itself, or by presenting some sort of credentials to the source, as though it were a "normal" user. These options will be explained in more detail later.

Secondly, there must be some way of restricting the search results to only those people authorized to see them. There are several ways we might do this, with advantages and disadvantages to each.

Enabling Security with an Identity Manager

Before Oracle Secure Enterprise Search may be used in secure mode, it must be linked to an Identity Manager.

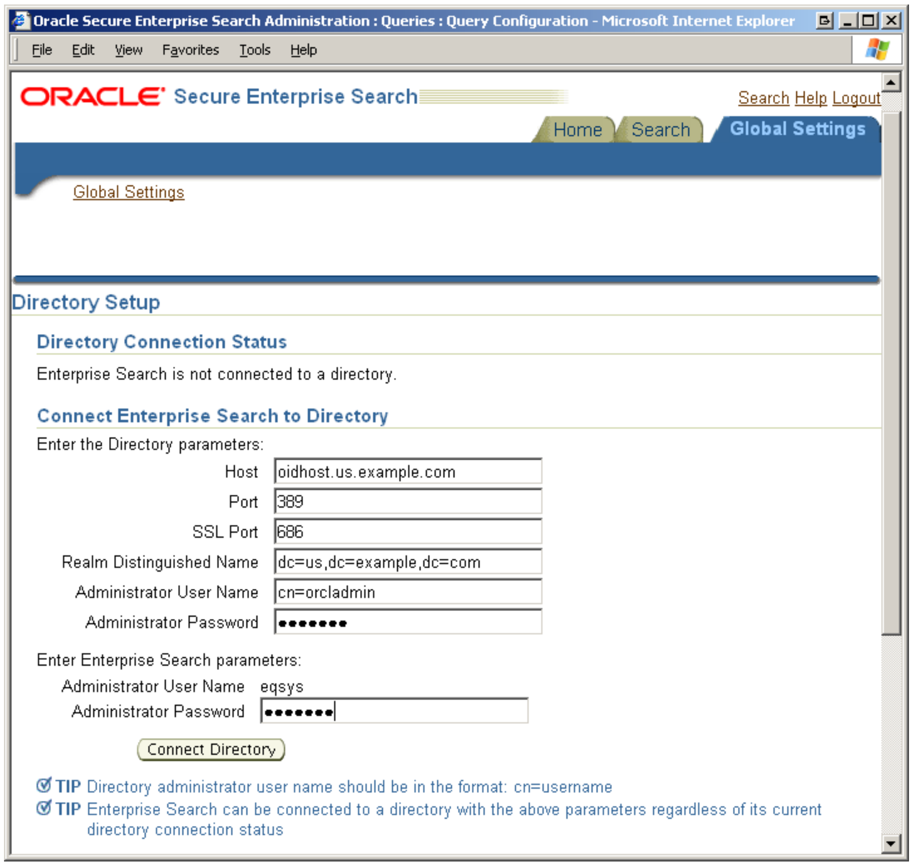
Oracle provides Identity Managers for many common LDAP-based directory systems, such as Microsoft Active Directory (AD) and Oracle Internet Directory (OID). In addition, it provides identity managers based on the internal user stores for many document management systems and other data sources, and of course an API so you can add your own identity managers if required.

Initially, after linking with an identity manager, users will have the option of searching in unsecured mode, or logging in via a form-based login screen in Oracle SES itself.

Automatic log in (often known as "Single Sign On" or SSO) is supported for users of Oracle's OID and Microsoft Active Directory using the Windows Native Authentication (WNA) feature based on the Kerberos protocol.

Single Sign On using Oracle Internet Directory

Once Oracle SES is configured to use SSO in OID, users need only login once to seamlessly perform searches then jump to the repositories containing search results.


Oracle Secure Enterprise Search Administration : Queries : Query Configuration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
ORACLE Secure Enterprise Search
Search Help Logout
Home Search Global Settings
Global Settings

Directory Setup

Directory Connection Status

Enterprise Search is not connected to a directory.

Connect Enterprise Search to Directory

Enter the Directory parameters:

Host	oidhost.us.example.com
Port	389
SSL Port	686
Realm Distinguished Name	dc=us,dc=example,dc=com
Administrator User Name	cn=orcladmin
Administrator Password	●●●●●●●

Enter Enterprise Search parameters:

Administrator User Name	eqsys
Administrator Password	●●●●●●●

TIP Directory administrator user name should be in the format: cn=username

TIP Enterprise Search can be connected to a directory with the above parameters regardless of its current directory connection status

SSO can be configured so that users must authenticate before performing any searches, or else so that unauthenticated users can perform searches against public sources, but private sources are only made available once the user is logged in.

In SSO mode, all calls to the search application are passed via Oracle AS. AS then checks the client's SSO authentication. If the user is already logged into SSO, AS will connect to Oracle SES via a secure HTTP connection, and will reroute communications to the client. If the user is not logged in, and authentication is compulsory, AS will pass the client to the SSO login page, and once the client is authenticated, will reroute communications to Oracle SES as above. In this mode, SES will only communicate with AS, so it is impossible for a client to bypass SSO login and access the search directly. If unauthenticated searching is allowed, then users will be routed directly to the search application until they choose to log in.

Note: Oracle Internet Directory and Single Sign-On are components of Oracle Application Server. Oracle Application Server must be licensed and installed independently from Oracle SES.

Single Sign On using Windows Native Authentication

When using Microsoft WNA with Kerberos, users simply sign into Windows and are automatically logged into SES and other supported applications without any further login dialog being required.

Securing Search Results

Here we look at three mechanisms for the search to identify users authorized to see documents.

Option 1 – No Checking

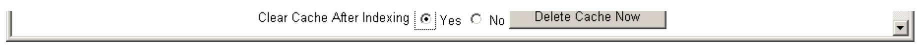
The simplest method, applicable when your documents are public or unsecured, is not to worry about checking at all at query time. We can present the list of results, with a list of pointers (normally URLs) to the original document. In this approach, if the document is protected by the application that supplies it, then users can see the document in their hitlist, but when they come to click on the link, the link fails, or they get a message saying they are not authorized to read that document.

This approach has some advantages. Firstly, it's simple. Oracle SES doesn't need to do any kind of security checking and there is no need for search users to log in before searching.

Secondly, it may actually be acceptable in some cases for the search to show "unauthorized" results. For example we might imagine a news source that has some subscriber-only content. We might want to show these documents in the hitlist, and then prompt the user to subscribe when they try to view the document. Alternatively, we might just want to alert them to some content that they could then apply for permission to see.

However this approach is unacceptable in high-security situations. Just knowing that documents exist with certain search terms may breach the confidentiality model. And the user would be able to discover more and more about the document in question by carefully refining their search.

Note that in this model it is important to disable the caching of documents locally. Otherwise users can just view the cached version of the document to bypass security. This can be done from the Administration tool by going to Global Settings / Configuration and setting: "Clear Cache After Indexing" to "Yes" (*note:* as a global setting this applies to all sources)


Clear Cache After Indexing [✓ Yes] No

Option 2 – Query Time Authorization

In this model (known as "late binding") we do not apply any kind of security restrictions to the query itself, but then while building the hitlist, before displaying it to a user, we call a custom Java class to check whether the user has access rights to each document in turn. The Java class will typically call some function in the original application to check for rights. If the rights check succeeds, we put the document into the hitlist, if it fails we leave it out.

The advantage of this last-moment check is that it will never show unauthorized documents, and the security information is guaranteed to be up-to-date.

The disadvantage is that it can be slower, especially if the user is only authorized to view a small subset of the total documents, as most of the hits will need to be pruned from the hitlist, and we may need to check thousands of documents before we can find enough to fill our 10-item hitlist.

Users will always be required to login for this model, as otherwise the search application has no way of knowing which user is performing the search.

Option 3 – ACL Checking

In this model (known as "early binding")we store access information alongside each document that we index. This is normally done by storing an Access Control List, or ACL, for each document. An ACL (pronounced variously as *ay-cee-el* or *ackel*) is a fragment of XML which describes which users or groups of users have access to the document. Access includes privileges to read, write or delete the document, though for our purposes we are only interested in read access.
Normally, an ACL is applied to a whole datasource, that is: all documents in a particular datasource will have the same ACL (or set of ACLs). However, in the case of a user defined datasource it is possible to for the datasource to supply document-level ACLs – each document can have its own specific ACL.

ACL checking is generally fast – it can be included as part of the query itself, which is much more efficient than per-document checking. However, it has the disadvantage that it is only stored at crawl time, so if the access permissions for a document have changed since the last crawl, this won't get reflected, and we may get a "false hit" or miss that document, depending on how the permissions have changed. For a false hit, the user would not normally be able to access the original document, since the current access control would be enforced by the document source, but if a cached version of the document is available the user would be able to see that.

Option 4 – Combine ACL Checking with Query Time Authorization

It is also possible to combine options 2 and 3 ("early and late binding"), such that ACLs are used to pre-filter the list of hits down to a manageable size, and then each hit is checked individually against the original source to make sure those documents are still accessible by this user. This would improve performance over Option 2, and prevent false hits where the ACL has become more restrictive. You would, however, still miss hits where the ACL has become *less* restrictive. In general it is not a security risk to have "false negatives", so this option is the most secure.

For example a document management system might have many thousands of users, each with their own small set of viewable documents. To run all of these documents through Query Time Authorization would be expensive – you'd typically need to prune perhaps 99.9% of the documents resulting from any search. So by applying ACLs from the source, we can ensure that a search is correctly targeted at the small set of documents viewable by the search user. We can then use QTA as a "secondary filter", to make sure that the user doesn't see any documents which have been restricted since the crawl took place (perhaps, due to an oversight, some documents were inadvertently made public then later restricted).

Note: If users are granted access to additional documents, these will not become available to them for searching until the next time a crawl is run.

Identity-Based Security versus Attribute-Based Security

SES Security can work in one of two modes - Identity-Based or Attribute-Based (sometimes described as User-Defined) Security.

Identity Based Security

Identity-Based security is the simpler of the two options. Here, a document is visible to any combination of *users* and *groups*. So a document may be tagged with an ACL which says it is accessible to the users A, B and C, and also to any user who is a member of the groups D, E and F.

In this case, the Identity Manager is responsible for resolving group membership. When a user logs in, the Identity Manager is called to get a list of the groups of which that user is a member. This information is used to construct a Security Filter which is part of every query to be run. Effectively a query will look like:

"find all documents containing *searchterm* which are accessible to user *username* or users in groups *group1, group2, group3*"

Attribute Based Security

This option relies is useful where the security model is more complex than simple users and groups, or where group membership is defined in the source system rather than the identity manager.

In this case, certain attributes of the source are defined as *Security Attributes* (there can be *grant* and *deny* attributes, but we'll just consider grant attributes for now).

The format of a security attribute is entirely up to the source, but it typically consists of a list of short strings. A document is only visible to a user when the security attributes in the source match the security attributes for a user.

For every source that uses security attributes, there must be an associated *Authorization Manager* plugin.

In the same way that the Identity Manager is responsible for returning the groups of which a user is a member, so the Authorization Manager is responsible for returning the security attributes for a particular user. When the user logs in, all available authorization managers are called to get the security attributes for the current user, and these security attributes are used (alongside user and group info from the identity manager) to construct the security filter.

Note that a source may define more than one security attribute - such as *role* and *responsibility* for example. If this is a case then there must be a match on both (or all) of the attributes - it is not sufficient just to get a match on one of the attributes. If this is not the desired functionality, then a single security attribute should be used, and the values for that attribute coded in some manner to show which type of value they are - for example we might use "ROLE_A" and "RESP_B" to distinguish the two types.

Secure Crawling Options

Before displaying the Search results as outlined above, the Search engine has had to build an index by crawling the various sources you want to search. There are a number of ways of implementing crawls securely. The Oracle Secure Enterprise Search Crawler has to be authenticated in order to be able to access the appropriate data sources. If the data source rejects external agents like crawlers, then they will not be searchable. If data sources allow all external agents access to their contents, then they are not secure. It is therefore important that the authentication of the crawler allows a way to search content without compromising security.

We will be looking at several methods of authentication:

1. Admin-based Authentication

2. Service-to-Service Authentication
3. Self-Service Authentication
4. Custom Crawler
5. Federated Search
6. Public Information

Note that option 5 does not involve a crawler, since responsibility for the search – and therefore for all aspects of security – are passed to the data source application.

Admin-based Authentication

This is the simplest form of crawler authentication. In this scenario, the Oracle SES administrator defines a datasource, and specifies the login credentials to be used for authorization within that datasource.

The actual method used for authentication will depend on the datasource type. For example when defining an email datasource, the username and password will be those for the IMAP user whose email we wish to index (note that Self Service Authentication is usually a better choice for email – see later).

If we are providing authentication for a web datasource, there are three methods of providing admin-based authentication:

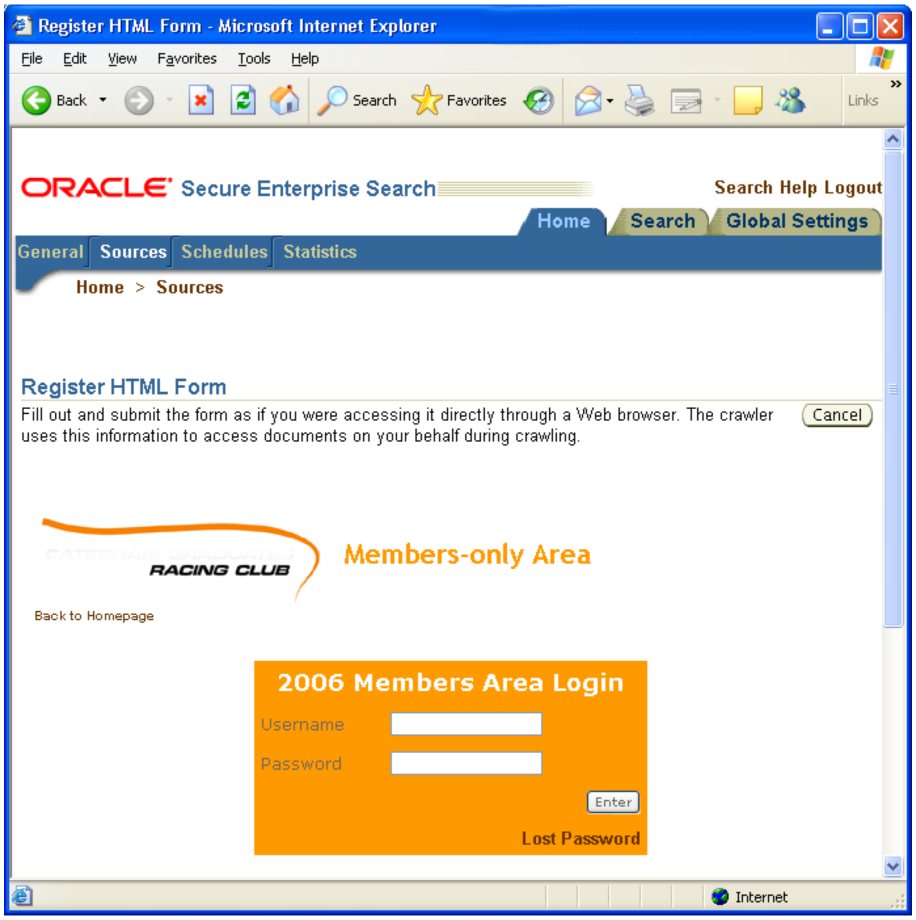
- HTTP Basic Authentication
- Form-based Authentication
- Oracle Single Sign On Authentication

Basic Authentication

HTTP Basic Authentication is a standard method of authentication built into the HTTP protocol. Prompting for credentials is normally performed by the user browser (typically by popping-up a username/password box), and the credentials are passed across to the server as parameters to each HTTP call

Form-based Authentication

Form-based authentication is probably more common in commercial websites, as it gives web designers more scope to customize the login experience, and is often perceived as more friendly than the stark login prompt of HTTP Basic Authentication. However, given the many varieties and layouts of login forms, it is more complex task to specify how the SES crawler should perform the login. For simple HTTP forms, Oracle SES provides a wizard, which will monitor the web page in question as you login, and collect the necessary information for logging in at crawl time. For more complex (or JavaScript based) forms, the admin user must provide the needed information manually.


Register HTML Form - Microsoft Internet Explorer

Register HTML Form

Fill out and submit the form as if you were accessing it directly through a Web browser. The crawler uses this information to access documents on your behalf during crawling.

File Edit View Favorites Tools Help
Back Search Favorites Links
ORACLE Secure Enterprise Search
Search Help Logout
Home Search Global Settings
General Sources Schedules Statistics
Home > Sources
Cancel
RACING CLUB
Members-only Area
Back to Homepage
2006 Members Area Login
Username
Password
Enter
Lost Password
Internet

Single Sign On Authentication

If your application is protected by Oracle Single Sign On, the admin user need only provide a simple login and password.

A single datasource may contain multiple protected areas, each using its own login details. For Basic Authentication, we can specify the domain and realm for a set of credentials. We can also define multiple forms, and the crawler will use the supplied login details as and when it finds the specified form. The nature of Oracle Single Sign On means that only one set of credentials is supplied, and this is applied to all SSO-protected sites encountered. If different SSO credentials should be supplied for some site, this should be indexed from a separate datasource.

In most cases with admin-based authentication, the administrator will need to set up multiple datasources for different users and/or groups of users. So the administrator is assuming responsibility not only for *enforcing* a security policy, but also for *defining* it. These multiple datasources may be combined, if required, via the scheduling mechanism within Oracle SES so they all get crawled together.

Storage of Usernames and Passwords

Normally, usernames and password supplied via the admin interface are stored in a secure, encrypted format in a digital "wallet" in a hardened Oracle database embedded inside the Oracle SES engine. Once stored, these credentials can used again and again to crawl the data sources under desired schedules.

In some cases the long-term storage of login credentials may be disallowed by local security policies. In such cases, you can use the option to "Delete Passwords After Crawl". In this case, automatic regular scheduling of the datasource will be disabled. The schedule must be launched manually from the admin interface, and the password supplied at launch time. The password will be retained in Oracle Secure Enterprise Search temporarily, for the duration of the crawl only.

Service-to-Service Authentication

In service-to-service (or *S2S*) authentication, the datasource treats the SES crawler as a *trusted application*. This means it will provide the crawler with any information it requests, along with ACLs, on the understanding that Oracle SES will enforce the ACLs and prevent access to information by unauthorized users.

To implement an S2S crawler, the data provider must support Access Manager Authentication Web Service (a protocol defined and owned by Sun Microsystems, but available on a variety of platforms).

To run through the acronyms, this uses the Simple Authentication and Security Layer (SASL) to add authentication support to the Simple Object Application Protocol (SOAP) transport layer. More information can be found at

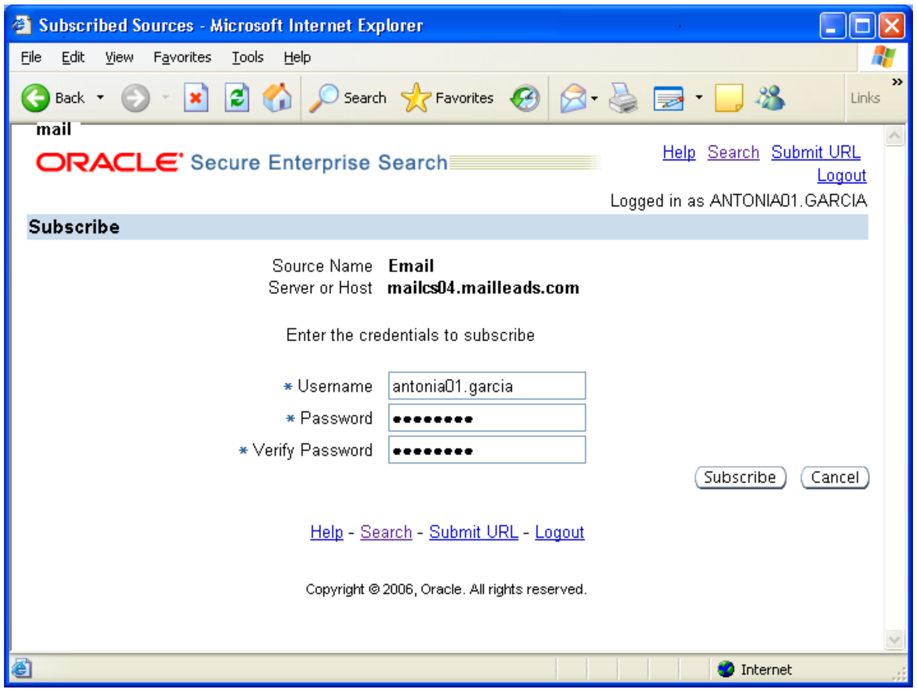
http://docs.sun.com/source/817-7648/authn_web.html

Self Service Authentication

In Admin-based Authentication, described above, the administrative user must know and provide all access credentials. This would frequently be considered undesirable. In Self Service Authentication, only the owner of the data needs to know these details.

Self Service Authentication is a two or three step process:

1. The admin user creates a template data source, that is, a definition of the type of information, its location, etc.
2. The ordinary user logs into the Oracle SES query application, and selects the Self Service option. He is presented with a list of the available template data sources. He chooses the sources, or sources he wishes to be crawled, and enters his login credentials on an input form. These credentials are then stored encrypted in the Oracle SES embedded database, in a secure wallet.
3. A schedule will be created for this user's crawl, but will not yet be launched. If the schedule was launched immediately, there would be a risk that many users would start indexing sources during peak hours. Instead, the administrator will be responsible for launching the newly created schedule at a suitable time.


Subscribed Sources - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Links
mail
ORACLE Secure Enterprise Search
Help Search Submit URL
Logout

Logged in as ANTONIA01.GARCIA

Subscribe

Source Name **Email**
Server or Host **mailcs04.mailleads.com**

Enter the credentials to subscribe

* Username	antonia01.garcia
* Password	••••••••
* Verify Password	••••••••

Subscribe Cancel
Help - Search - Submit URL - Logout
Copyright © 2006, Oracle. All rights reserved.
Internet

Self service authentication is also useful for datasources that do not provide an admin based authentication or service to service authentication. This might be relevant in the case of IMAP servers and external websites.

Custom Crawler Plug-in API

A custom crawler may be written in Java to provide access to resources not crawled by any of the standard crawler types provided with Oracle SES.

The custom crawler plug-in returns documents to the indexing engine. Each document can be accompanied by an ACL, which restricts the users who may access that particular document. Alternatively, an ACL can be set through the admin screens for the whole datasource.

Authentication credentials can be provided, if needed, to the plugin via parameters, passed through from the admin screens. The plug-in will use these credentials to access the secure source.

An example of a secure crawler plug-in can be downloaded from the SES web page on Oracle Technology Network (OTN), at http://www.oracle.com/products/technology/oses.

Most of the connectors to the third-party information systems are implemented as custom crawlers, and therefore inherit the characteristics of this model.

Federated Search

Oracle SES also supports a “federated search” model.. This means that part, or all, of the search is carried out by an external entity. That entity may be another SES instance, or it may be a completely separate application.

In terms of secure search, we are most interested in separate applications. For example, an email system (such as Oracle Beehive Email) may contain its own text indexes. In this case, we may decide that rather than letting Oracle SES crawl the email data, we should let Beehive do the searching for us. So when a user enters a query, rather than executing it locally, we pass it across to Beehive for execution, and collect the results. The results are then merged with any results from other data sources, which are searched at the same time, and presented in the normal SES hitlist manner.

Federated Search has several advantages:

1. No crawling is necessary, freeing up resources and space on the Oracle SES system.
2. Depending on the search characteristics of the target system, the search is more likely to reflect recent new and changed documents. A normal SES search is only as up-to-date as the last crawl, whereas other systems may have real-time index update capabilities.
3. There is no need for any local storage of user credentials in wallets. Security is handled by the target application, which will only return hits which are viewable by the logged-in user.

Against this, we have the disadvantages that performance may be lower, due to the increased network traffic, and possibly slower search performance on the target machine.

The federation model fully supports secure search, by means of "Federation Trusted Entities". These can be considered as special system usernames, protected by either a simple password, or an entry in the connected identity manager. In order for a federation "broker" to access a federation "end point" in a secure manner, it must know the entity name and associated password defined on the end point. In this way, a trust relationship is established between broker and endpoint, and the broker is able to run queries on the endpoint on behalf of a particular user, without needing to provide the user's actual login credentials (which may not be available to the broker, especially if a single-sign-on system is in use).

User Authorization Cache

The User Authorization Cache is a performance enhancement to improve the fetching of group information from the identity manager. It is optional, but can serve to both reduce the load on authorization servers and increase the performance of SES queries.

Previously, the list of groups for which the user is a member was derived when the user logged in, and then periodically refreshed before running a query once a certain timeout period has expired.

In some circumstances (notably when Active Directory is in use and there are many nested group definitions in use) this group lookup can take some considerable time, and can slow down logins and queries after the timeout period. In some cases, the group lookup can timeout completely.

SES version 11g introduces the "User Crawler" source type. This crawler type does not fetch information for indexing, but instead collects group information for users.

This group information is then cached, and made available when the user logs on or runs queries, without the need to look the information up in the identity manager.

Creating a user crawler source causes the creation of a schedule, like any other source. This schedule can be run manually, or (more usefully) set to run periodically.

The benefits of this approach are:

- The fetch of group information is asynchronous, so does not slow down the process of login, or the running of queries

- The schedule can be set to run at a quiet time of day, thus avoiding excessive loads on the directory system at peak times.

A user authorization cache can be created for any identity manager type.

Note that the user crawler needs to be able to request a list of users from the identity manager, functionality which was not previously available from identity managers. Therefore a new interface "CrawlableIdentityPluginManager" has been introduced to the SES Java API. Several supplied identity managers have been extended to support this interface. User written identity plugins will need to be modified to support the new caching function. The user crawler schedule will fail if it is run against a non-compliant identity manager.

In future versions, it is intended to extend this user authorization cache to include security attributes from authorization plugins (for when attribute based security is in use), but in the initial release of 10.1.8.4 the cache stores only group information from the identity manager.

Securing the Search Index

Since it contains content and ACL information from secure sources, your search index must be more secure and harder than your secure sources. Oracle SES stores the search index not on a file on disk that could be compromised, but inside a specially hardened database embedded inside SES. This database is not available for external access.

By utilizing the Oracle database in this manner we get advantages from the thousands of man-years of development work. Oracle SES is

- Unbreakable – Oracle SES is built on the unbreakable Oracle 10g platform, which has more security certifications than any other vendor.
- Fully Globalized – Unicode and all other database character sets are supported, and documents are supported in more than 100 languages.
- Available on multiple platforms – Windows and Linux initially, followed by other Unix implementations.
- Scalable and Performant – built on the Oracle Text engine which has been proven to handle many terabytes of information in actual customer implementations.

CACHE FILE STORAGE

In SES 10g, cache files (HTML renderings of the indexed documents) were stored on the file system of the SES server. In 11g they are stored more securely in the database itself, using the Oracle 11g Secure Files feature. When moving from SES 10g to SES 11g, administrators can choose on a source-by-source basis whether to store cache files in the file system or in the database. There may be advantages to choose file storage for some less-secure storage – for example the administrator may wish to store such cache files on slower, cheaper storage devices than those used for the main SES database files.

Searchable Sources

Oracle SES can securely search all your sources. Searchable repositories include:

- HTML pages served up by a Web Server.

- Database Tables - Search Oracle databases and any other databases that support the ODBC standard. Database tables can reside in Enterprise Search's own database instance, or be part of a remote database accessed over a network. Both full text columns and fielded columns can be crawled.
- Files - Local or remote files can be made searchable through the file:// protocol.
- Emails - Emails and mailing lists can be crawled over the IMAP protocol. Additionally there is support for Microsoft Exchange and Lotus Notes Email.
- Oracle10g Application Server Portal repositories – including private and public Portal pages, folders, subfolders and text items.
- Desktop content via integration with desktop search engines such as the Google Desktop for Enterprise, which can be optionally configured to search locally downloaded email, files on local storage, previously viewed web pages etc.
- Many document management systems such as Lotus Notes, Documentum, OpenText Livelink and others
- Application systems such as Siebel and Oracle E-Business Suite
- Other sources – the Secure Crawler Plug-in SDK allows customizing the crawler to access other repositories.

Not all types of authorization (search security) and authentication (crawler login) capabilities are appropriate to all types of source. For example, Form login is only applicable to Web sources, and per-document ACLs (Source ACLs) can only be provided by sources which actually attach access rights to each document.

The following chart shows which authorization and authentication capabilities are available with each source type in Oracle SES.

	Authorization			Authentication				
Source Type	Source ACL	Supplied ACL	QTA	Form Login	SSO	Self Service	Source Login	Service to Service
Web		X	X	X	X			
Table		X	X		X			
File		X	X					
Email		X	X			X	X	
Mailing List		X	X			X	X	
Portal	X	X	X		X			
Federated	X	X	X		X			X

Custom	X	X	X				X	

Systems not listed above are generally implemented as Custom sources, and the details from the Custom line should be assumed.

Conclusion

No search engine can claim to be an "enterprise" search engine unless it is able to index protected sources, and return search results from those sources in a secure manner. Oracle Secure Enterprise Search provides a variety of techniques to suit many different applications and many different security requirements.

Oracle is committed to developing practices and products that help protect the environment

White Paper Title
July 2011
Author: Roger Ford
Contributing Authors: Jinyu Wang, Stefan Buchta

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together