



IMPLEMENTING SECURE ENTERPRISE SEARCH



WHITE PAPER

Shankar V Sawant
May 27, 2009

Enterprises have realized the importance of a search engine in exploring their internal hidden treasures and are convinced of its ROI, but the security and access control of the information exposed through a search engine still remains the most challenging and critical part of the solution. Although the problem is common for all enterprises, the implementation often requires a custom approach for each organization.

This white paper discusses various factors associated with implementing a secure search solution. The white paper further discusses what level of support the major enterprise search vendors provide to implement secure enterprise search.

Table of Contents

1. WHAT IS THE CONCERN?	2
2. ENTERPRISE SEARCH SECURITY DEFINED	2
3. SEARCH SOLUTION VULNERABILITIES	2
4. IMPLEMENTATION APPROACH	3
5. VENDOR SUPPORT FOR THE SEARCH SECURITY APPROACHES	5
6. GUIDELINES FOR IMPLEMENTING SECURE SEARCH	6
7. CONCLUSION	6

1. What is the concern?

Today's enterprises have their information and data stored in structured as well as non-structured format in various network locations. With the growth of the organization, non-structured data grows and organizations easily loose track of the information.

When any such organization is ready to implement a search engine one of the questions that concerns them the most is: how can I make sure only authorized users find secure information?. Search engine, if implemented carelessly has potential to expose proprietary, restricted content or in some cases verify the existence of hidden information to the unauthorized parties. The consequences of such scenarios can be very serious to the business.

2. Enterprise Search security defined

Search engine security is primarily a form of access control mechanism, which ensures that users can only retrieve information they are permitted to see.

Search engine security can be applied and managed at various levels of granularity. It is important to identify the level of granularity required for a specific implementation. Followings are the granularity levels that can be considered:

Collection level or Repository Level - This is simplest level of granularity in which access control is applied at collection index level or repository level. For example separate collection indexes can be created for public content, intranet content, vendor/partner content, department content and access control information then can be mapped easily between users/groups and respective collections. The administrator can configure these collections individually thereby allowing multiple users/groups access to these collections.

Document level - Both public and private (or secure) content can reside in the same collection index. In this case, documents in the index are tagged with "Access Control Properties". In majority of the implementations, access control is applied at this level of granularity.

Field Level - This is a finer level of granularity than document level, in which only a certain part of the document is indexed and shown in the result. It can be best implemented with structured documents like XMLs, DB etc. In this case access control is applied only to the part of document that is tagged with a predefined fields or tags.

Sub field Level - This is another, even more advanced level of granularity, in which specific terms and references will be removed, whilst still allowing partial disclosure. Moreover most of the search engines can be configured to crawl or exclude the URLs inside the document.

3. Search Solution Vulnerabilities

A search engine has two main tasks: one is crawling & indexing (process of collecting information about documents and creating indexed collection), the other is content serving (result display). Search solutions can prove vulnerable to security threats at either of these stages.

Following are some of the security holes that an incorrect implementation of enterprise search can expose.

Crawling and indexing sensitive information - This is the most critical security implication of a careless search implementation. A thorough planning and design, combined with 360 degree testing is required to minimize the risk.

Full path and/or metadata disclosure - This can expose the internal structure of the repository and can give malicious users an entry into the sensitive area.

Informative search result - Even though actual documents are secured by access control, a search engine normally indexes all the documents and if search results expose secure documents in the result list (with title and/or summary) with the intent to ask for credential later, it may prove a security threat as the title and/or summary may provide or confirm the sensitive information.

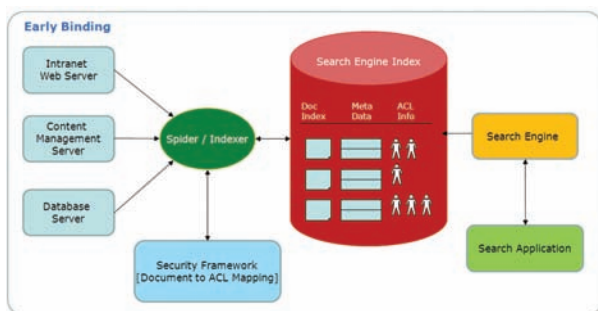
Security credentials caching - Some search engine vendors provide a solution that enables caching of security credentials. If security credentials are not synchronized regularly with the master system, un-authorized users can get access to the secure information as the search engine will not consult the main security provider while serving up the content.

4. Implementation approach

Like any enterprise system, implementation of search requires careful analysis of the existing infrastructure, complete requirements gathering and a roadmap of the implementation. Beyond regular planning, security policy analysis and monitoring is of paramount importance.

Although implementations are vendor-specific, there are primarily two distinct approaches used for filtering restricted (private) content from final result set: early binding - filtering of private information using security information collected at Crawl and Index phase and late binding - filtering of private information by checking the credential directly with underlying security systems during result serving phase. In most of the scenarios a hybrid approach is followed to get 'best of both the world' benefits. The Section below explains these approaches in more detail.

Early binding

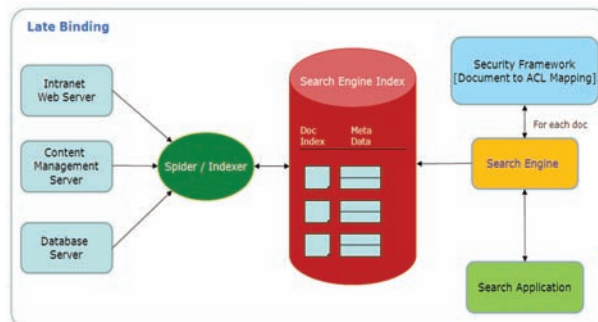


In this approach the search engine collects all the security and access control related information during crawling & indexing and stores it with the index (the actual implementation approach may vary by vendor and platform). In a way, the search engine tries to mimic the security framework locally. So when a user makes a query, the search engine attaches the user's credentials to the query so that the search engine fetches information that the user is entitled to see. This is like using a "where" clause in an SQL query while querying the database to fetch the filtered result-set. Similar to the database scenario, the search engine knows the access control information about the document beforehand. Compared to late binding, early binding security is often more complex to set up, because it is difficult to model all the security policies of the various back-end sources in the index and implement the comparison logic in uniform way. More over some of the vendors do not support synchronization out of the box but require re-indexing or delta indexing to update the security information.

Since early binding mirrors the security and access control framework to some extent, synchronization with the underlying security system is very important. On

the bright side, this approach yields a faster and seamless search experience. The best fit business scenario for early binding is where most of the content is in database or managed by tools like content management system, because these systems have very structured access control mechanisms that can be utilized by search engine to replicate the access control information locally.

Late binding



In this approach, the query is executed in a generic manner to fetch all the matching results from the collection and just before forming the final result set, the search engine checks to see if the access control flag on the document is "public" or "private" and in case the flag is private the search engine consults the respective underlying security systems to see whether the user has access to that particular document or not. The external system responds with "Yes" or "No" and that decides whether to include the document in the final result set. In the worst case where there is no provision for setting such flag, identifying the document as "public" or "private", the results list formatter will check every matching document against an external server to see if the user has access. Late binding document filtering can potentially be very slow and can strain corporate security systems, because each underlying security system will add its latency during credential check.

To overcome obvious implications on the performance, vendors of search engines provide various caching mechanisms and also parallel processing options.

Hybrid approach

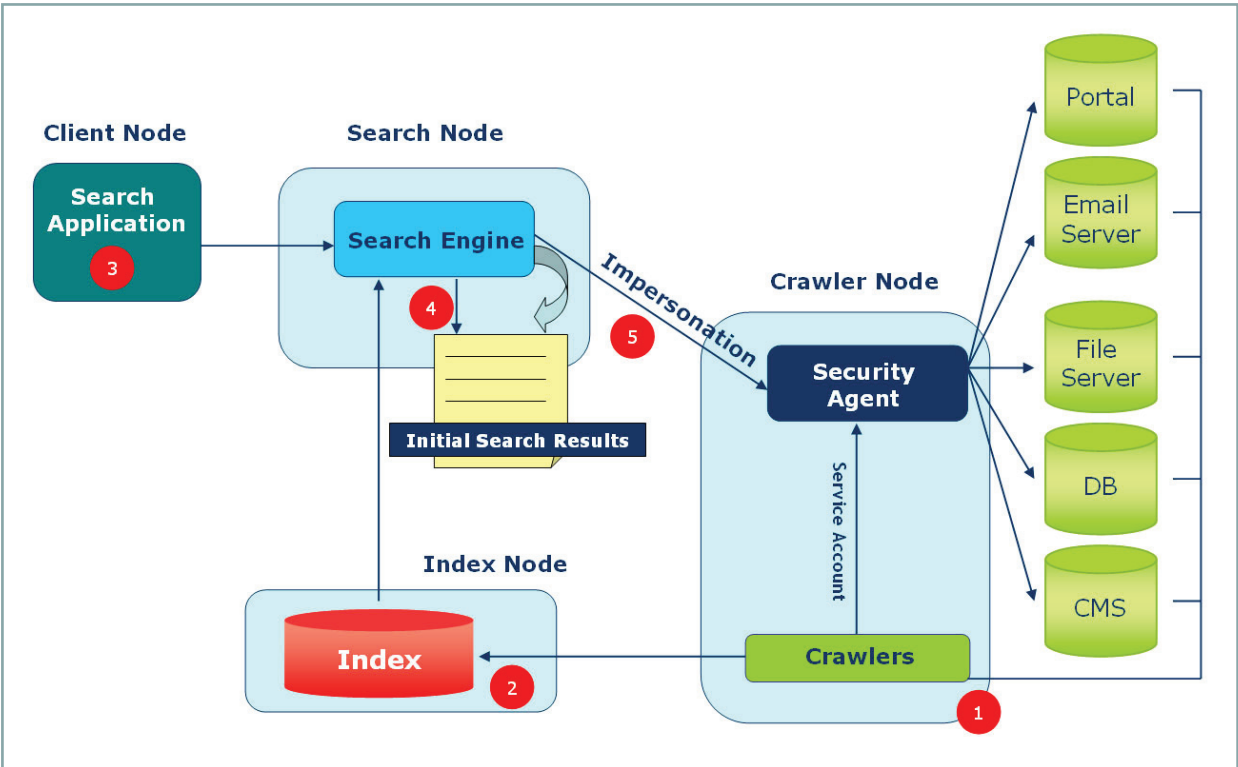
This approach, if supported by a search engine vendor, is more efficient than the individual approaches described above in that it combines the benefits of early and late binding. One of the options is to collect access control information of less frequently changing documents and use early binding filtering with them. For frequently changing documents, use simple flagging as "Secure" or "Public" during "Crawl & Index" phase and then use the late binding filtering to see if user has access to those documents during result formatting.

Sample Hybrid Implementation approach

In most of the practical scenarios there will always be some kind of hybrid implementation approach. This section discusses a possible implementation. The goal is to provide an optimal performance whilst maintaining the precise security policies of the originating document repositories. By storing high-level access control data in the index (like if the document is "public"/"private"/"Departmental" etc), the system can provide an interim (potentially smaller) result set that can then be post-filtered to verify current access controls. That way search engine is neither duplicating the complete security infrastructure nor iterating through large global result set but verifying the credential to smaller interim subset of most likely final result set.

The diagram below illustrates the implementation approach. The high level steps involved are as follows

- | | |
|--|--|
| 1 Collect Native Security Information | 4 Search Engine to produce Interim Results |
| 2 Store Security token with document in Index | 5 Post-filter with impersonation |
| 3 Create User's Credential context object | |



1. Extract native or high level access control information at crawl time.
2. Store access control information in the index.
3. Create the user's security context when the user logs in or when the session is initialized.
4. Process the search with the user's security context and produce an interim result set that contains only those documents that the user has access to at the repository level (or some other criteria can be used to create an interim result).
5. Filter the interim result set by consulting the back-end sources that contributed documents to the result set for current native ACL information. The decision to consult the back-end or not can be conditional based on a set of properties or custom logic using APIs provided by the search engine.

Advanced security considerations

Besides standard security factors discussed above, enterprises may need to consider the following implementation scenarios and the security challenges related to them in implanting a search solution. The details about these topics though are out of scope for this paper.

Security with Federated search

In Federated search, search results from various search engines (or more broadly information retrieval systems) are combined together to present user with a final result-set based on his/her credentials.

The implementation of federated search dictates the security or access control of the solution.

Implementing search security can be effective if not simple in federated search in that if individual search engines/Information retrieval systems manage the access level at their end and return the filtered result-set. Also in federated search mode there is no need to create a master service account having "Full" access to create a master Index, individual departments can maintain their Indexes and secure it natively.

Single Sign On

Search engines use "service account" while crawling and indexing backend systems but while serving the result, user might access a link or document, which requires logins. Single Sign On, if implemented can present a seamless user experience. Most of the leading search engine provides OTB support or provides SPIs to create single sign on solution. In practical scenario few documents may still require re-login.

Contextual Search

Contextual search allows users to search for a any particular term or topic in a searched document without leaving the context of the original document. From the security perspective, user's credentials need to verified against the ACL of the new contextual search results.

5. Vendor Support for the Search Security Approaches

1. Autonomy Intelligent Data Operating Layer (IDOL)

Result filtering options - Autonomy IDOL supports both early and late binding approaches. Early binding is known as mapped security; late binding is known as un-mapped security. Moreover IDOL supports a hybrid approach to provide best-of-both-worlds benefits.

Core security engine - Autonomy's early binding security model maps the underlying security model ACL, group, role, protective markings etc from all the underlying repositories directly inside the kernel of the

IDOL engine and stores the mapping in an encrypted format. This enables Autonomy IDOL search engine to serve search results based on the user's entitlements without interacting with the enterprise security systems in real time. To keep the security credentials in synch with the underlying repositories, Autonomy implements a transitional signaling mechanism within the connector layer to get the updates about the changes in the permissions for the indexed content. The effect of this is that whenever there is change in the underlying system with respect to permission/access control, IDOL updates those documents with latest information.

2. Google Search Appliance (GSA)

Result filtering options - GSA employs a late binding approach when implementing secure search. During crawling & indexing, GSA flags the content as public or private and during the content serving phase it verifies the user's access to content, only when that content is flagged as private.

Core security engine - During crawling & indexing, Google Search Appliance provides all the standard mechanisms like HTTP Basic or NTLM HTTP option for authentication. Along with the basic options GSA can be configured for Kerberos or Integrated windows authentication.

During content serving, GSA takes a two-step approach, the first step establishes the identity of the user requesting the search result and in the second step GSA impersonates the user and performs an authorization check on behalf of that user.

GSA provides SAML Authentication and Authorization Service Provider Interface (SPI) for integrating with existing security infrastructure.

3. IBM OmniFind

Result filtering options - With IBM OmniFind, it is possible to implement early binding as well as late binding search security strategies.

Core Security Engine - OmniFind can be used to set up document level security by configuring the crawler to associate a security token with document they crawl. These tokens are then stored in the index along with the document and when a user searches for these documents, OmniFind matches the user's credentials with the document tokens to decide whether to include that document in the search result or not.

However, these security tokens can get out of synch very easily as OmniFind does not provide any native method to update the credentials when they get changed in the underlying system. Custom plug-ins have to be written to implement this synchronization.

4. FAST Enterprise Search Platform (ESP)

Result filtering options - FAST supports both early and late binding implementation approaches and also both can be configured together to form a hybrid approach

Core Security Engine - FAST ESP delegates security mapping to Security access module which includes 'ACL monitor' for maintaining ACL information about indexed document and 'User monitor' for users and group information. For implementing more stringent security FAST provides APIs.

5. Solr

Solr, an open source enterprise search engine project based on the Lucene search APIs does not provide any out-of-the-box feature to filter documents based on access control. However it does provide an API for implementing this feature.

6. Guidelines for implementing secure search

1. Gather and analyze the security related requirement for an Enterprise Search.
2. Work with the corporate security team to understand the security policies. Identify and define security policies.
3. Decide on the level of granularity in implementing security viz. collection level, document level etc
4. Take extra precaution to avoid crawling and indexing information that should never be shown in the search results like configuration files, contracts or financial documents. Use exclusion patterns and settings in vendor-provided administration console to avoid crawling and indexing this critical information.
5. Implement stringent access control policies for the service account created for the crawlers to crawl the data sources. Generally this service account has to be given global access, so special attention is required. A digital certificate could be used to implement secure crawling.
6. For simple access control, show all results; let the security system ask for passwords (i.e. late binding).
7. For hit-level access control, Check with security system before displaying search results (late binding). Make sure disallowed results are never shown.
8. Implement a hybrid binding approach to balance between performance and security.
9. Secure various consoles like Administrative GUI, Analytics GUI etc. Implement a delegated administration option to delegate sub-set of admin responsibility to other admin users.
10. Beware of cached data
11. For highest level security protect server hardware and software, limit and log access to the search interface, encrypt transmissions during indexing and serving results.
12. For pages protected during transit by encryption (SSL), the search engine indexer can use an SSL client for access. The server then needs to be protected as much as the original server, and to serve results pages encrypted to avoid unauthorized access in transit
13. When implementing federated search, the security credential check should be performed at the individual search engine level.
14. Implement search engine usage analytics controls (provided by vendors) to monitor and analyze the search engine usage and use the findings to continuously improve the search engine.
15. Last but not the least develop a comprehensive test strategy and perform all security testing including third party penetration testing.

7. Conclusion

Almost all implementations of enterprise search involve some form of security requirements. The security implications and the approaches discussed above will form the basis for the solution. Every organization will have its own security policies and challenges and search engine security will be an addition. But the careful planning, implementation, testing and then monitoring the use from security perspective will help the organization to implement a secure enterprise search solution.

Contact us

USA

460 Park Avenue South
Suite #1101, New York
NY 10016, USA
Tel.: +1 212 686 6655
Fax: +1 212 686 2422

UK

88 Wood Street
London EC2V 7RS,, UK
Tel.: +44 20 85281000
Fax: +44 20 85281001

Australia

410 Concord Road
Rhodes, NSW 2138, AUS
Tel.: +61 290 221 146,
Fax: +61 290 221 134

INDIA

Bagmane Technology Park
Byrasandra Village,
C.V. Raman Nagar
Bangalore 560 093, India
Tel.: +91 80 4004 0404,
Fax: +91 80 4004 9999

About Mphasis

Mphasis is a leading Applications, Infrastructure Technology, and BPO services provider.

The company delivers real improvements in business performance for clients through a combination of technology know-how, domain and process expertise. With currently over 36,000 people, Mphasis services clients in Financial Services, Healthcare, Communications, Media & Entertainment, Transportation & Logistics, Energy & Utilities, Consumer & Retail, and Governments around the world. To know more, visit www.mphasis.com.

Mphasis and the Mphasis logo are registered trademarks of Mphasis Corporation. All other brand or product names are trademarks or registered marks of their respective owners. Mphasis is an equal opportunity employer and values the diversity of its people. Copyright © Mphasis Corporation. All rights reserved.

