Lotus Developer Domain
The Technical Resource for Lotus software
LDD Today

Level: Intermediate
Works with: QuickPlace
Updated: 01-Apr-2003

by
Joe
Russo

Have you ever had an authentication or authorization failure when using QuickPlace and been unable to get the information you need to resolve the problem? Well, this article is intended to open the black box of information and to give you insights into how QuickPlace authentication really works. This is the first of two articles that look closely at QuickPlace authentication. Part one discusses the following authentication-related areas:

- Basic Authentication
- Single sign-on (SSO) authentication
- QuickPlace server settings, particularly security and user directory settings

These articles shows you many of the ways you can troubleshoot your QuickPlace authentication problems. You'll find this valuable for two reasons: to debug and understand your configuration, and to give you an insight into the architecture of QuickPlace authentication. This article is intended for experienced QuickPlace system administrators. An understanding of Domino administration, LDAP directories, and XML is helpful.

Authentication

QuickPlace supports both Basic Authentication and multiple server single sign-on (SSO) as well as customization of authentication through the use of DSAPI (Domino Server API), but that is outside the scope of this article. For the purposes of this article, we look closely at Basic Authentication and single sign-on and how to troubleshoot problems with both.

Basic Authentication

Basic Authentication is a standard access control method supported by Web browsers. This method is so basic that there is no protection of the credentials being passed between client and server, so to ensure that no user names or passwords are intercepted, you need to SSL-enable the server. SSL (Secure Socket Layer) is discussed later in this article.

First, let's understand what happens during Basic Authentication. A user enters the URL for his/her Place. The browser issues the request for this URL, and it sends no user name or password information—in effect, the request is anonymous. Let's assume that this Place is *not* open for anonymous access. So, the server tries to field this request anonymously, finds that it cannot, and lets the browser know that credentials are required. The browser presents the user with a dialog box asking for user name and password. The user enters these, and the credentials are sent to the server along with the request (again). If this is accepted by the server, then for all subsequent requests to the same realm, this browser resends the user name and password, which are encoded in the HTTP header in base 64.

Now, if this user is a member of another Place, when entering the URL for that place, the user is asked to authenticate again even if he is using the same browser session. This happens because when the browser issues

the request for this new Place, it does *not* send the user name and password because the realm for this new Place is not currently known to the browser. The realm for the first Place may be http://server.com/firstplacename, and this new Place may be http://server.com/secondplacename. So the browser issues this request as anonymous, and the same steps occur as previously described.

You can set Basic Authentication following these steps:

1. With your Notes client, open the names.nsf database on your QuickPlace server.
2. In the Server folder, open the Servers view, select the Internet Protocols Tab, and select Domino Web Engine.
3. Under the HTTP Sessions heading, set the Session Authentication field to Disabled.

You are now using Basic Authentication.

User cache, what's that for?

Let's take a moment to discuss the user cache and what it means. As we noted above, the browser sends the user name and password on all requests to the Place. This requires that the authentication and authorization be processed for each and every request. This is not very efficient. To optimize this functionality, the server employs a user cache, so that when a user first authenticates with a Place, the authentication information is stored in a cache so that on subsequent requests, the cache information can be reused.

Let's get into the details of what happens in the user cache using an example. Consider what happens to a new user entering the system and what happens when that user re-enters the system.

User Kyle Russo wants to go to his QuickPlace, SoccerGoons. He enters the URL for this QuickPlace in his browser and is prompted to log in. He enters his user name and password. Now, the authentication code first checks the user cache for this user.

Checking the cache

Here is how QuickPlace checks the cache step-by-step.

1. QuickPlace begins by getting the first user in the cache.
2. If there is one, then we can continue.
3. QuickPlace looks at this user name to see if it has expired. To check for expiration, QuickPlace compares the current time with the expiration time stamp on this user's cache entry. If this user is older than the expiration time stamp, QuickPlace removes this user from the cache.
4. If this user has *not* expired, then QuickPlace checks for a match. How QuickPlace matches depends on the authentication model. If it performs Basic Authentication, then it matches on the user name as entered along with the password. If QuickPlace uses Mutli-Server Session Authentication, it merely matches on the name token, which in this case is the distinguished name (DN)—a unique name. QuickPlace cannot match on password in this case because it does not have one.
5. In this example, QuickPlace doesn't find a match with the first entry, so it proceeds to the next user entry in the cache. It repeats Step 3.
6. After running through all the members in the cache, QuickPlace finds that Kyle is not here.

So now the code proceeds by peforming the actual authentication of this user and obtains a DN. After all group membership for this user's DN is obtained, QuickPlace has the User Credential Information. Now, QuickPlace adds this user to the user cache.

Adding a user to the cache

When QuickPlace adds a user to the cache, it:

1. Sets the timestamp on this user cache entry.
2. If the cache is at or above its limit, QuickPlace removes the entry at the bottom of the list.
3. Puts this entry at the top of the user cache list.

Now, let's imagine some time has passed, and Kyle makes a request on the server. Here's what happens: The browser sends the request along with Kyle's authentication information. If this is Basic Authentication, then the original user name/password pair is sent; if this is Session Authentication (multi-server), then the session cookie is sent. So now, as above, QuickPlace checks the user cache.

1. It begins again by getting the first user in the cache.
2. Because QuickPlace knows that there is at least one, it continues.
3. It looks at this user name to see if it has expired. To check for expiration, QuickPlace compares the current time with the expiration time stamp on this user's cache entry. If this user is older than the expiration time stamp, QuickPlace removes this user from the cache. Now, if this user who expired was Kyle, it's as if he never had a cache entry, and the authentication/authorization process must occur again. Though this may

seem bad, QuickPlace needs this in order to remove stale cache entries. There needs to be a mechanism in which changes to a user's entry, either password (infrequent), user name (rare), or group membership (most likely) can be reflected in the system without having to start/stop the server. The default two minute user cache timeout is really overkill for this kind of need however.

4. If this user has *not* expired, then QuickPlace checks for a match. How QuickPlace matches depends on the authentication model. If it performs Basic Authentication, then it matches on the user name as entered along with the password. If QuickPlace uses Mutli-Server Session Authentication, it merely matches on the name token, which again is the DN. Remember QuickPlace cannot match on password, in this case, because there isn't one.
5. In this example, QuickPlace finds a match, so it moves this user cache entry to the top of the list and re-sets this user entry time stamp.
6. QuickPlace stops looping over the user cache entries because it found a hit. It returns the User Credential Information from the cache, and Kyle can continue on with his desired operation.

Now what?
So now we know the dirty details of caching of users, what can you do with this information? Suppose you want the following:

- The fastest possible access for users to system resources
- Cached entries used as much as possible (a pure state of nirvana would be to always use a cached entry except for the first time during some user's session)
- Cached entries kept up-to-date (to reflect the current state of the directory for this user)—which means we want to remove stale entries as soon as they are stale

To accomplish these goals, consider these items:

- To do what you want, you must make use of a user cache.
- To make timeouts as large as possible (or to never time out), you must make the cache size as large as possible.

Time and space
Or as Einstein would say, time-space. First, consider time. A "never time out" setting is really a poor choice because it fills the user cache with stale entries and clearly violates the third goal of keeping cache entries up-to-date. Instead, you want to determine a setting that reflects a typical user session, something longer than two minutes. Experiment with this setting in your environment.

Second, consider space—the final frontier. You dictate size in terms of entries in the cache. However, in real terms, computers deal in bytes of memory, so you need some rule of thumb to tie one to the other. Then you can determine a setting for how many users should be cached based on a computer's memory capacity. A user cache entry is made up of a user’s distinguished name and all of the distinguished names of their groups, let’s say that a typical one is about 1000 bytes. So a user cache limit of 100 would require about 100,000 bytes (or about 97 Kbytes) of data (1,000 would require about one MB). Additionally, you should tune the user cache to the capacity experienced during a typical session. For instance, if you set the cache timeout to 20 minutes, look at the maximum number of users you expect during any one 20 minute window. That tells you what size you need and whether or not you have the system memory to handle it. If you cannot supply the necessary memory, then you need to reduce the session timeout until you get to a reasonable number (or buy a system that can handle the use it is being subjected to).

In QuickPlace, you can change settings for the user cache. In Notes.ini, you can add two settings: QuickPlaceMaxCachedUsers and QuickPlaceExpireCachedUsers. QuickPlaceMaxCachedUsers is the setting used for the number of entries you need in the user cache. By default, QuickPlace uses the settings that are in the Domino server record for Maximum Cached Users. The default is 64.

QuickPlaceExpireCachedUsers is the setting for the cache timeout in seconds. Again, QuickPlace reads the value from the Domino server record; the default is 120 seconds.

What is a distinguished name?
A distinguished name (DN) is a string composed of name components that uniquely addresses some entity in a directory. In QuickPlace, we are interested in people and groups. When a person or a group is added to a Place from the user directory, it must contain the distinguished name for this person or group. The DN is added to the appropriate areas in the Place. The DN is the token which QuickPlace uses to identify a person or group for specific access. All other data items for a person or group are for UI and contact purposes; the DN is the token used to identify that person or group. When you log in, your DN and the DNs of your groups are compiled into a list. This list of DNs is passed to the QuickPlace server commands to grant or deny you access.

When you authenticate with a Place, there are three important steps that take place:

- The user name and password are passed to the authenticator for verification.
- Once verified, the user name and password pair become a Distinguished Name (DN), which is a unique name for the user.
- This DN is then used to find your group membership; each group that you are a member of also has a DN.

The user cache information holds the DN for the user and all the DNs for the groups that you are a member of. In the Server document under the Memory Caches heading, the setting for Maximum Cached Users is the same setting used for the User Cache size, and the setting for Cache User Expiration Interval is the timeout value for users in this cache. These fields determine the user cache for QuickPlace.

When using Basic Authentication, the requests are always sent with the user name and password pair, so this is part of the key by which the user cache is searched to produce the cache entry data. Furthermore, because QuickPlaces have groups that are local to the Place itself, the user cache entry *must* exist on a per user, per Place basis. When a request comes into the server, QuickPlace uses the user name, password, and Place name to search the cache to find the matching entry.

What if you're having problems with Basic Authentication and you need to debug?

Basic Authentication troubleshooting

First, inspect the HTTP headers. You can use a tool like the IBM Page Detailer to see all of the HTTP headers being sent between client and server.

You'll see something like this:

WWW-Authenticate: Basic realm="SomeRealm"

The HTTP header passes the user name and password information. For more information about the HTTP header, see the **RFC:2617 HTTP Authentication: Basic and Digest Access Authentication**.

The browser sends the user name and password separated by a colon character (:) as a base64 encoded string.

Authorization: Basic TGVlIFJ1c3NvOnNvY2NlcnBsYXllcg==

You can use base 64 decoding to retrieve the value. You should be able to find a reasonably useful tool on the Internet. Search on the Web "base 64 decoder free." Here are a couple of tools to check:

- **Web-based Base64 Converter**
- **BASE64/RADIX64 Coder**

(By the way, for those of you keeping score at home, this decodes to Lee Russo:soccerplayer.)

Single sign-on

In single sign-on, the user is presented with a form login to enter user name and password. This form, when submitted, is processed by the Domino login command. The login command takes the user name and password and calls into the authentication component to verify the user. If successful, it results in a distinguished name of the verified user. This DN is encrypted using the LTPA key and made into a string. The servers sends this string or token to the browser as a cookie. On subsequent requests from the browser, the cookie is sent, and the server decrypts the cookie to determine the user. The decrypted cookie produces the Distinguished Name of the user.

LTPA (Lightweight Third Party Authentication) is an IBM standard. Domino and WebSphere servers can share the LTPA public/private key to encrypt/decrypt distinguished names that can then be trusted by all servers that share the same LTPA public/private key pair. A DN must be unique within that directory.

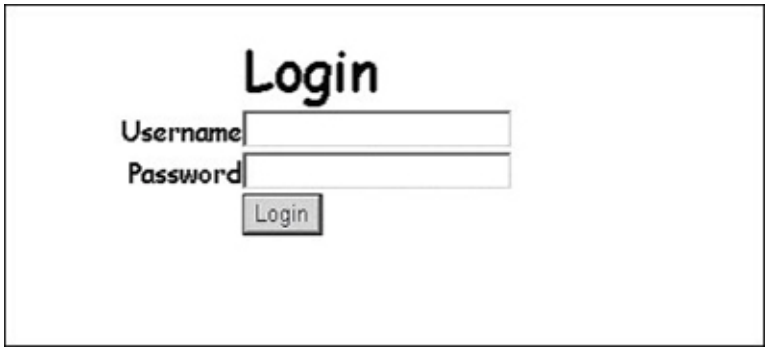
The Domino single sign-on form is a form that is part of a special database. In the absence of the database, the Domino server presents a "built-in" form. You can also create one of these databases using the template database DA50.NTF. The default form in this case is identical to the built-in one. For QuickPlace, the default form presents a problem. The Domino form assumes that the Domino server is the gatekeeper for authentication. As such, it is hard-coded to direct the authentication to the names.nsf database. For QuickPlace, this does not work.

On a QuickPlace server, there can be many QuickPlaces, each of which are the gatekeepers to authenticated access to their databases. To work around this problem, QuickPlace has a form in the resources.nsf database. This version of the login form uses a JavaScript redirection to ensure that QuickPlace requests work as well as

Domino database requests. This database is known as domcfg.nsf. When using the Domino default form, the browser shows the title Server Login. When using the QuickPlace form, the entire background is white, and the title is simply Login. It is a very common mistake that QuickPlace LTPA authentication fails because the wrong login form is being used.

When the wrong form is used, you see the following screen:

When the right form is used, you see the following screen:



Here are the steps needed to enable LTPA for the Domino server that hosts QuickPlace.

1. In your Notes client, open names.nsf database, select the Server folder, and then open the Servers view.
2. Select the Server document and open it in edit mode.
3. Select the Internet Protocols Tab, then select the Domino Web Engine Tab.
4. Under the Session Authentication heading, select Multiple Server.
5. Save the Server document.
6. While still in the Servers view, click the Web Configurations action.
7. Select Web SSO Configuration.
8. Enter the correct domain name and server name. Then enter the session timeout.
9. Finally, you need to get or make the public/private key pair (SSO keys).
10. After you do this, save the Configuration document and close the names.nsf database.

Refer to the **Domino Administrator help** for more details on how to setup Session-based authentication.

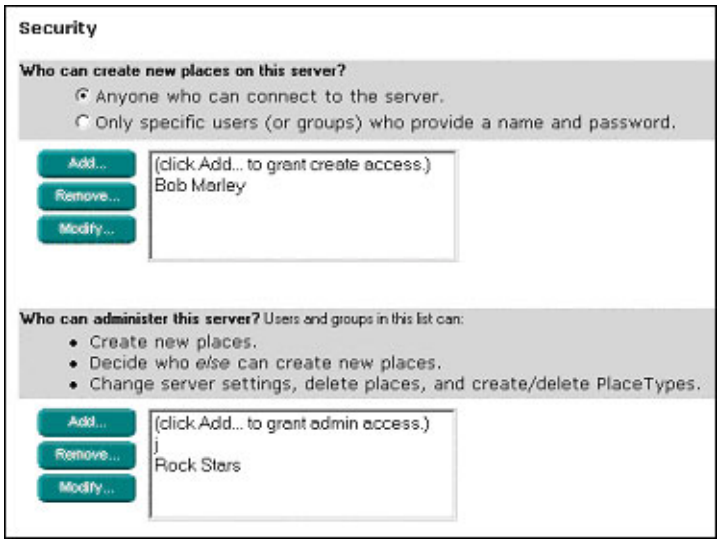
Server settings

In the Administration QuickPlace, there is a room for Server Settings. Let's examine those that are important to authentication.

Security

This area has two settings:

- Who can create new places on this server?
- Who can administer this server?



First, you can open the Create QuickPlace room so that anonymous users can access this room and create Places. When you set this option for anonymous users, any user who connects to the QuickPlace sees the Create A QuickPlace table of contents item. If you set the “Only specific users who provide a name and password” option, then only those users who you add as Creators or Administrators see the Create A QuickPlace table of contents item.

Second, you can select a list of users who can create Places. Finally, you can select those users who have administration privileges.

User Directory
In this section, you can:

- Connect the QuickPlace server to an external user directory for user authentication and group membership
- Enable or disable the ability to create users specifically for use within a single Place
  The membership and password information for these users is maintained totally within the confines of the Place where they are created. These users are known as Local users.

The preferred method of connection for the user directory is the Lightweight Directory Access Protocol (LDAP). This is a standard for directory access connections. Let’s explore some of the QuickPlace settings for LDAP directories.

Name
This is the DNS name (or IP address) of the LDAP Server.

Base
When Place managers lookup users to add to their Places, they use the Lookup UI, which issues an LDAP search. This setting indicates the base hierarchy to search within the LDAP directory; this is known as the Search Base Distinguished Name (see later in this article for more information on DNs). An LDAP directory is organized as a tree with an organization or country typically used as the root. Then there can be levels of organizational units below this root, finally with a name or unique identifier for each individual user or group. The QuickPlace search base setting lets you limit all lookups to this base setting. Some LDAP directory implementations require a search base to be used, while others do not require a base (meaning that when searching, the entire LDAP directory is searched).

Port
This is the port number on which the LDAP directory is configured to accept LDAP queries. You should check with your LDAP directory administrator to get the correct setting. Default for non-SSL LDAP is port 389.

Search with Credentials
This section allows you to enable or disable the use of credentials for searching. Check with your LDAP directory administrator to see if the LDAP directory allows anonymous searches (in which case, this setting is not needed). If anonymous searches are not allowed on your LDAP directory, then you need to check this setting and to enter the user name and password that enables anonymous search. Whenever a Place manager needs to lookup a user in the LDAP directory, these credentials are used to bind to the LDAP directory to perform the search.

SSL
The SSL setting enables SSL encryption for communications between the QuickPlace server and the LDAP server. If the Domino/QuickPlace server has not cross-certified with the LDAP server, then this setting results in LDAP communication failures.

The process for engaging SSL for communication between the QuickPlace server and the LDAP server is as follows:

1. Enable SSL on the Domino/QuickPlace server. The Domino/QuickPlace server exchange public keys with the LDAP server. Refer to the **Domino Administrator help** for the detailed steps on how to enable SSL.
2. Cross-certify the Domino/QuickPlace and LDAP servers, then test the connection to ensure that the Domino server and the LDAP server can communicate via SSL.
3. The easiest way to do this is to set up a connection in the Domino server to this LDAP server using Directory Assistance. Note that you do NOT need a Directory Assistance database to enable the QuickPlace LDAP user directory; this is merely a troubleshooting step.
4. Create a new Directory Assistance database using the Directory Assistance database template.
5. After you create the database, add a new DA record, selecting LDAP as the protocol. Under the rules tab, select YES for trusted credentials. Under the <UNKNOWN> tab, enter the DNS name for your LDAP server. Initially, do *not* select SSL for the connection—one thing at a time. Save this record; you are prompted by a warning dialog box telling you that you have a non-secure connection, but click OK and ignore it.
6. Add a setting to your Notes.ini to tell Domino to trust authentications with the LDAP server. The setting is:

   INET_AUTHENTICATE_WITH_SECONDARY=1

7. Also add the following debug setting to show what's going on with the LDAP communications:

   WEBAUTH_VERBOSE_TRACE=1

   Note that both of these settings are *only* for this test, so make sure to remove them when you go into production mode because they decrease performance.
8. Create another database using the Public Discussion database template, call it TestDB.nsf. In this database, open the access control list and do the following:
   - Set the Default access to No Access.
   - Add a user to the ACL, one whose distinguished name and password you know from the LDAP directory. You should add him/her using the Domino Abbreviated DN form. For example, suppose your directory has a DN for user Lee Russo. His DN may be cn=Lee Russo, ou=United States, o=FIFA. In the ACL, enter Lee Russo/United States/FIFA. Give the user Editor access.
   - Save the database and close it.
   - Start your Domino server, making sure you run the HTTP task.
   - After the HTTP task is up and running, open a Web browser and enter the URL http://yourserver/TestDB.nsf.
   - A prompt box should ask you for user name and password. Enter the user name and password for

the user you added to the ACL. If you have a good connection with the LDAP directory, you should be recognized as Lee Russo/United States/FIFA.

At this point, if you are successful, continue to the SSL test. If this test fails, then the LDAP authentication connection is not working. With the logging output on the server console, you should be able to determine why the authentication failed. After you correct this problem and pass through the last step, continue to the next procedure.

1. Stop the server.
2. Edit the Directory Assistance database record for the LDAP server, and change the setting to enable SSL.
3. Save the record.
4. Start the Domino server again, making sure to run the HTTP task.
5. After the HTTP task is up and running, open a Web browser and enter the URL http://yourserver/TestDB.nsf.
6. Again, you should get a prompt box, asking you for your user name and password. Enter the user name and password for the user you added to the ACL. If you have a good connection with the LDAP directory, you should be recognized as Lee Russo/United States/FIFA.

At this point, if you have succeeded in step 6, then you have verified that the SSL connection is working between the Domino/QuickPlace Server and the LDAP Server. Open your browser and enter the URL for the Administration QuickPlace (http://yourserver/quickplace). Then open the Server Settings room, go to the user directory setting, and select the SSL checkbox.

If this step failed, but you were successful in the previous section, you have verified that you can connect to your LDAP Server, but you cannot do so over SSL. Double-check your SSL settings.

When QuickPlace is SSL-enabled, it uses the Domino SSL Handshake to handle the encryption on all QuickPlace to LDAP calls.

Timeouts

QuickPlace uses the LDAP directory for two functions. QuickPlace calls to the LDAP directory to authenticate users (except for Local users). The authentication timeout value is used whenever an authentication attempt is made. If the authentication attempt cannot be made within the timeout specified, then the authentication fails with a timeout error.

QuickPlace also uses the directory to search for users and groups. The search timeout value is used whenever one of these searches is made, and if the search results cannot be returned in the time specified, then the search fails with a timeout error. This is a useful setting if the LDAP directory is rather large and you want to prevent excessive processing of results. For example, if a Place manager selects the wildcard (*) search, it could take a considerable amount of time and memory to present the results. The timeout setting prevents this from impacting the QuickPlace server performance by cutting short this request.

Conclusion

This article introduced you to QuickPlace's handling of Basic Authentication and single sign-on and explained how to troubleshoot problems with both security methods. It also looked at the QuickPlace security and user directory settings which may help you to diagnose authentication problems. In the second part of the series, we look at QuickPlace as a DSAPI server and application; user directory customization and configuration; and authentication, authorization, and LDAP debugging.

ABOUT THE AUTHOR

Joe Russo is the Administration Audience lead developer for QuickPlace 3.0. He has a seldom used Electrical Engineering degree from SUNY at Buffalo where he graduated too many years ago to mention. He spent the early part of his career in the imaging and graphics world and then hopped onto the Internet development bandwagon. He lives with his wife and three sons somewhere in Massachusetts, in seclusion from his admiring fans. He hopes one day to write a book that people (other than friends and relations) would be willing to purchase.

