HttpAuthenticationSchemes

Introduction

This is a feature in Nutch that allows the crawler to authenticate itself to websites requiring NTLM, Basic or Digest authentication. Work and information to support POST based authentication that depends on cookies can be found at: HttpPostAuthentication

Necessity

There were two plugins already present, viz. 'protocol-http' and 'protocol-httpclient'. However, 'protocol-http' could not support HTTP 1.1, HTTPS and NTLM, Basic and Digest authentication schemes. 'protocol-httpclient' supported HTTPS and had code for NTLM authentication but the NTLM authentication didn't work due to a bug. Some portions of 'protocol-httpclient' were re-written to solve these problems, provide additional features like authentication support for proxy server and better inline documentation for the properties to be used to configure authentication.

JIRA NUTCH-559

These features were submitted as JIRA NUTCH-559 in the JIRA. If you have checked out the latest Nutch trunk, you don't need to apply the patches. These features were included in the Nutch subversion repository in revision #608972

Introduction to Authentication Scope

Different credentials for different authentication scopes can be configured in 'conf/httpclient-auth.xml'. If a set of credentials is configured for a particular authentication scope (i.e. particular host, port number, realm and/or scheme), then that set of credentials would be sent only to pages falling under the specified authentication scope.

When authentication is required to fetch a resource from a web-server, the authentication-scope is determined from the host and port obtained from the URL of the page. If it matches any 'authscope' in this configuration file, then the 'credentials' for that 'authscope' is used for authentication.

Configuration

Since the example and explanation provided as comments in 'conf/httpclient-auth.xml' is very brief, therefore this section would explain it in a little more detail. **In all the examples below, the root element <auth-configuration> has been omitted for the sake of clarity**.

Prerequisites

In order to use HTTP Authentication, the Nutch crawler must be configured to use 'protocol-httpclient' instead of the default 'protocol-http'. To do this copy 'plugin.includes' property from 'conf/nutch-default.xml' into 'conf/nutch-site.xml'. Replace 'protocol-http' with 'protocol-httpclient' in the value of the property. If you have made no other changes it should look as follows:

```
<property>
  <name>plugin.includes</name>
  <value>protocol-httpclient|urlfilter-regex|parse-(html|tika)|index-(basic|anchor)|scoring-opic|urlnormalizer-
(pass|regex|basic)</value>
  <description>Regular expression naming plugin directory names to
  include.  Any plugin not matching this expression is excluded.
  In any case you need at least include the nutch-extensionpoints plugin.
  In order to use HTTPS please enable
  protocol-httpclient, but be aware of possible intermittent problems with the
  underlying commons-httpclient library.
  </description>
</property>
```

Optional

By default Nutch uses credentials from 'conf/httpclient-auth.xml'. If you wish to use a different file, the file should be placed in the 'conf' directory and 'http.auth.file' property should be copied from 'conf/nutch-default.xml' into 'conf/nutch-site.xml' and then the file name in the '<value>' element should be edited accordingly. The default property appears as follows:

```
<property>
  <name>http.auth.file</name>
  <value>httpclient-auth.xml</value>
  <description>Authentication configuration file for 'protocol-httpclient' plugin.</description>
</property>
```

Crawling an Intranet with Default Authentication Scope

Let's say all pages of an intranet are protected by basic, digest or ntlm authentication and there is only one set of credentials to be used for all web pages in the intranet, then a configuration as described below is enough. This is also the simplest possible configuration possible for authentication schemes.

```
<credentials username="susam" password="masus">
 <default/>
</credentials>
```

The credentials specified above would be sent to any page requesting authentication. Though it is extremely simple, default authentication scope should be used with caution. This set of credentials would be sent to any web-page requesting for authentication and therefore, a malicious user can steal the credentials used in the configuration by setting up a web-page requiring Basic authentication. Therefore, we usually use credentials set apart for crawling only, so that even if a user steals the credentials, he wouldn't be able to do anything harmful. If you are sure, that all pages in the intranet use a particular authentication scheme, say, NTLM, then this situation can be improved a little in this manner.

```
<credentials username="susam" password="masus">
 <default scheme="ntlm"/>
</credentials>
```

Thus, this set of credentials would be sent to pages requesting NTLM authentication only. Now, one can not set up a page requiring Basic authentication and steal the credentials. NTLM is safer, because password is not sent in clear-text or in a form from which the original password can be recovered directly.

Credentials for Specific Authentication Scopes

The following is an example that shows how two sets of credentials have been defined for different authentication scopes. For all pages of example:8080 requiring authentication in the 'blogs' or 'wiki' realm, the first set of credentials would be used.

```
<credentials username="susam" password="masus">
  <authscope host="example" port="8080" realm="blogs"/>
  <authscope host="example" port="8080" realm="wiki"/>
</credentials>
<credentials username="admin" password="nimda">
  <default/>
</credentials>
```

However, an important thing to note here is that if some page of example:8080 requires authentication in another realm, say, 'mail', authentication would not be done even though the second set of credentials is defined as default. Of course this doesn't affect authentication for other web servers and the default authscope would be used for other web-servers. This problem occurs only for those web-servers which have authentication scopes defined for a few selected realms/schemes. This is discussed in next section.

Catch-all Authentication Scope for a Web Server

When one or more authentication scopes are defined for a particular web server (host:port), then the default credentials is ignored for that host:port combination. Therefore, a catch-all authentication scope to handle all other realms and scopes must be specified explicitly as shown below.

```
<credentials username="susam" password="masus">
  <authscope host="example" port="8080" realm="blogs"/>
  <authscope host="example" port="8080" realm="wiki"/>
</credentials>
<credentials username="admin" password="nimda">
  <default/>
  <authscope host="example" port="8080"/>
</credentials>
```

The last authscope tag for example:8080 acts as the catch all authentication scope. In this section, realms were used to demonstrate the example. The same holds true for schemes also. For example, in the following example, the last authscope tag is necessary if the second set of credentials must be used for all pages of example:8080 not belonging to the authentication scope defined in the first tag.

```
<credentials username="susam" password="masus">
  <authscope host="example" port="8080" realm="blogs" scheme="DIGEST"/>
</credentials>
<credentials username="admin" password="nimda">
  <default/>
  <authscope host="example" port="8080"/>
</credentials>
```

Important Points

1. For <authscope> tag, 'host' and 'port' attribute should always be specified. 'realm' and 'scheme' attributes may or may not be specified depending on your needs. If you are tempted to omit the 'host' and 'port' attribute, because you want the credentials to be used for any host and any port for that realm/scheme, please use the 'default' tag instead. That's what 'default' tag is meant for.
2. One authentication scope should not be defined twice as different <authscope> tags for different <credentials> tag. However, if this is done by mistake, the credentials for the last defined <authscope> tag would be used. This is because, the XML parsing code, reads the file from top to bottom and sets the credentials for authentication-scopes. If the same authentication scope is encountered once again, it will be overwritten with the new credentials. However, one should not rely on this behavior as this might change with further developments.
3. Do not define multiple authscope tags with the same host, port but different realms if the server requires NTLM authentication. This means there should not be multiple authscope tags with same host, port, scheme="NTLM" but different realms. If you are omitting the scheme attribute and the server requires NTLM authentication, then there should not be multiple tags with same host, port but different realms. This is discussed more in the next section.
4. If you are using NTLM scheme, you should also set the 'http.agent.host' property in conf/nutch-site.xml

A note on NTLM domains

NTLM does not use the concept of realms. Therefore, multiple realms for a web-server can not be defined as different authentication scopes for the same web-server requiring NTLM authentication. There should be exactly one authscope tag for NTLM scheme authentication scope for a particular web-server. The authentication domain should be specified as the value of the 'realm' attribute. NTLM authentication also requires the name of IP address of the host on which the crawler is running. Thus, 'http.agent.host' should be set properly.

Underlying HttpClient Library

'protocol-httpclient' is based on Jakarta Commons HttpClient. Some servers support multiple schemes for authenticating users. Given that only one scheme may be used at a time for authenticating, it must choose which scheme to use. To accomplish this, it uses an order of preference to select the correct authentication scheme. By default this order is: NTLM, Digest, Basic. For more information on the behavior during authentication, you might want to read the HttpClient Authentication Guide.

Troubleshooting

If you are having problems with your authentication configuration, it is a good idea to step back, start with a very basic configuration, keep testing it and gradually adding to it until you get your desired configuration working. At the very start, check that the account that your crawler is using is enabled and working on the server(s). To do this, try to access one of your test URLs with a web browser. When prompted, enter the details of your crawler's account. If this does not work, the problem is with the server and it will need to be fixed there.

The configuration below can be used as a starting point. It provides minimum detail, allowing the client and server maximum flexibility.

```
<auth-configuration>
  <credentials username="crawler-user-name" password="crawler-password">
    <default realm="domain" />
  </credentials>
</auth-configuration>
```

To check if your configuration is working, you can use the ParserChecker

It is easy to see whether it has fetched the page successfully even without looking into logs. If it is successful, it will display a proper page title and many links extracted from the page. Otherwise, it will display a title like “You are not authorized to view this page” and few links, if any.

If you look in the logs/hadoop.log file, search for the AuthChallengeProcessor records similar to this:

```
INFO  auth.AuthChallengeProcessor - ntlm authentication scheme selected
```

In case of failure, such a record will be followed by something like this:

```
INFO  httpclient.HttpMethodDirector - Failure authenticating ...
```

Need Help?

If you need help, please feel free to post your question to the user@nutch mailing list. The author of this work, Susam Pal, usually responds to mails related to authentication problems. The DEBUG logs may be required to troubleshoot the problem. You must enable the debug logging for 'protocol-httpclient' and Jakarta Commons HttpClient before running the crawler. To enable debug logging for 'protocol-httpclient' and HttpClient, open 'conf/log4j.properties' and add the following lines:

```
log4j.logger.org.apache.nutch.protocol.httpclient=DEBUG,cmdstdout
log4j.logger.org.apache.commons.httpclient.auth=DEBUG,cmdstdout
```

It would be good to check the following things before asking for help.

1. Have you overridden the 'plugin.includes' property of 'conf/nutch-default.xml' with 'conf/nutch-site.xml' and replaced 'protocol-http' with 'protocol-httpclient'?
2. If you patched Nutch 0.9 source code manually with this patch, did you build the project before running the crawler?
3. Have you configured 'conf/httpclient-auth.xml'?
4. Do you see Nutch trying to fetch the pages you were expecting in 'logs/hadoop.log'. You should see some logs like "fetching http://www.example.com/expectedpage.html" where the URL is the page you were expecting to be fetched. If you don't see such lines for the pages you were expecting, the error is outside the scope of this feature. This feature comes into action only when the crawler is fetching a page but the page requires authentication.
5. With debug logs enabled, check whether there are logs beginning with "Credentials" in 'logs/hadoop.log'. The lines would look like "Credentials - username someuser; set ...". For every entry in 'conf/httpclient-auth.xml' you should find a corresponding log. If they are absent, probably you haven't included 'plugin.includes'. In case you have manually patched Nutch 0.9 source code with the patch, this issue may be caused if you have not built the project.
6. Do you see logs like this: "auth.AuthChallengeProcessor - basic authentication scheme selected"? Instead of the word 'basic', you might see 'digest' or 'NTLM' depending on the scheme supported by the page being fetched? If you do not see it at all, probably the web server or the page being fetched does not require authentication. In that case, the crawler would not try to authenticate. If you were expecting an authentication for the page, probably something needs to be fixed at the server side.

Once you have checked the items listed above and you are still unable to fix the problem or confused about any point listed above, please mail the issue with the following information:

1. Version of Nutch you are running.
2. Complete code in 'conf/httpclient-auth.xml' file.
3. Relevant portion from 'logs/hadoop.log' file. If you are clueless, send the complete file.