HttpPostAuthentication

Introduction

Often, Nutch has to crawl websites with pages protected by authentication. Therefore, to crawl such web-pages, Nutch must authenticate itself to the website and then proceed with fetching the pages from it. Currently, the development version of Nutch can do Basic, Digest and NTLM based authentication. This is documented in HttpAuthenticationSchemes. In this project, we would be adding HTTP POST based authentication, which is the most popular form of authentication on most websites. It should be possible to configure different credentials for different websites.

Configuration

A configuration file with a list of domains for which authentication should be done along with the login URL and POST data. If possible, the configuration should also allow the user to mention a session timeout value for websites as an optional parameter. This would be helpful if some website is known to timeout very quickly, or when the duration of the fetch cycle would be too long as compared to the session's life.

Behavior of the fetcher

1. If the URL to be fetched is from a domain in the config file AND this is the first time the fetcher is going to hit the domain, it should first send the POST data to the login URL mentioned in the POST data configuration file for that domain and obtain the session cookies.
2. Save the cookies. (protocol-httpclient does this for a single fetch cycle, so this should be handled automatically).
3. Request the actual URL that was supposed to be fetched.
4. For further URLs from the same domain, authentication need not be done for the same fetch cycle.

However, there should be some exceptions to the last behavior, since a server session may expire the session, before our fetch cycle is complete. This would be a problem, if the crawler hits the website again later in a fetch cycle after the session has expired. Therefore, the exceptions to the last rule should be made if one of the following conditions is met:

1. If the URL redirects the page to the login URL (which can be checked from the configuration file) authentication should be done again.
2. If the URL returns an error page, authentication should be done again.
3. If the time elapsed after the last fetch from the website is more than the session timeout specified for the website, authentication should be done again.

Some challenges discussed in the mailing list

The authentication failure page may be returned as HTTP 200 OK status which makes it more difficult. Three possible ways to solve it:-

1. We use pattern matching to find out whether the contents of the page indicates it as an authentication failure page or not, for the website. But it is an unnecessary waste of time because for most cases the page wouldn't be an error page.
2. We perform an authentication by sending POST data to login URL every time we fetch a page from that domain. By this, we are almost doubling the bandwidth requirement to crawl that website.
3. For those sites, where authentication failure page comes from a known URL, we can add which URLs mean authentication failure along with the login URL and POST data in the configuration file. There wouldn't be too many such URLs for a particular domain and so a regex match or a complete string match for the URLs after every response from that domain shouldn't consume much time.

However, even without taking care of these points, and simply getting the fetcher behavior right as discussed in the previous section, we'll have a solution that may be useful to many.

Original discussion in the mailing list

http://www.mail-archive.com/nutch-user@lucene.apache.org/msg10248.html

Jira Issue(s)

NUTCH-827 - HTTP POST Authentication