Athena

A user's guide

By Steve Lord
(A friend of Dave's)
steve@buyukada.co.uk

”You are no longer a child: you must put childish thoughts away."
[Athena to Telemachus. Homer, *Odyssey* 1.296]

Background

Athena was the Greek goddess of wisdom, war, the arts, industry, justice and skill. She was the strongest supporter of Odysseus, and regularly helped both him and his son Telemachus throughout the Odyssey. Athena is also the name of a mighty fine Turkish band, but that’s not important right now.

I’d been working on a poorly written search engine query tool for a while when I heard about Foundstone’s SiteDigger. I decided to expand it into something better, using a modified (yet semi-compatible) form of the SiteDigger XML file format.

Requirements

Athena should work on any system with the .NET CLR runtime installed. So far it’s been tested on various Windows XP systems with .NET but 2000 should be fine. If anyone gets this working on Mono then I’d be very interested in hearing about it.

Further Information

The latest release of Athena is available from the Athena homepage an http://www.buyukada.co.uk/projects/athena/ - There is also an Athena mailing list accessible from the main Athena site.

Credit Where Credit’s Due

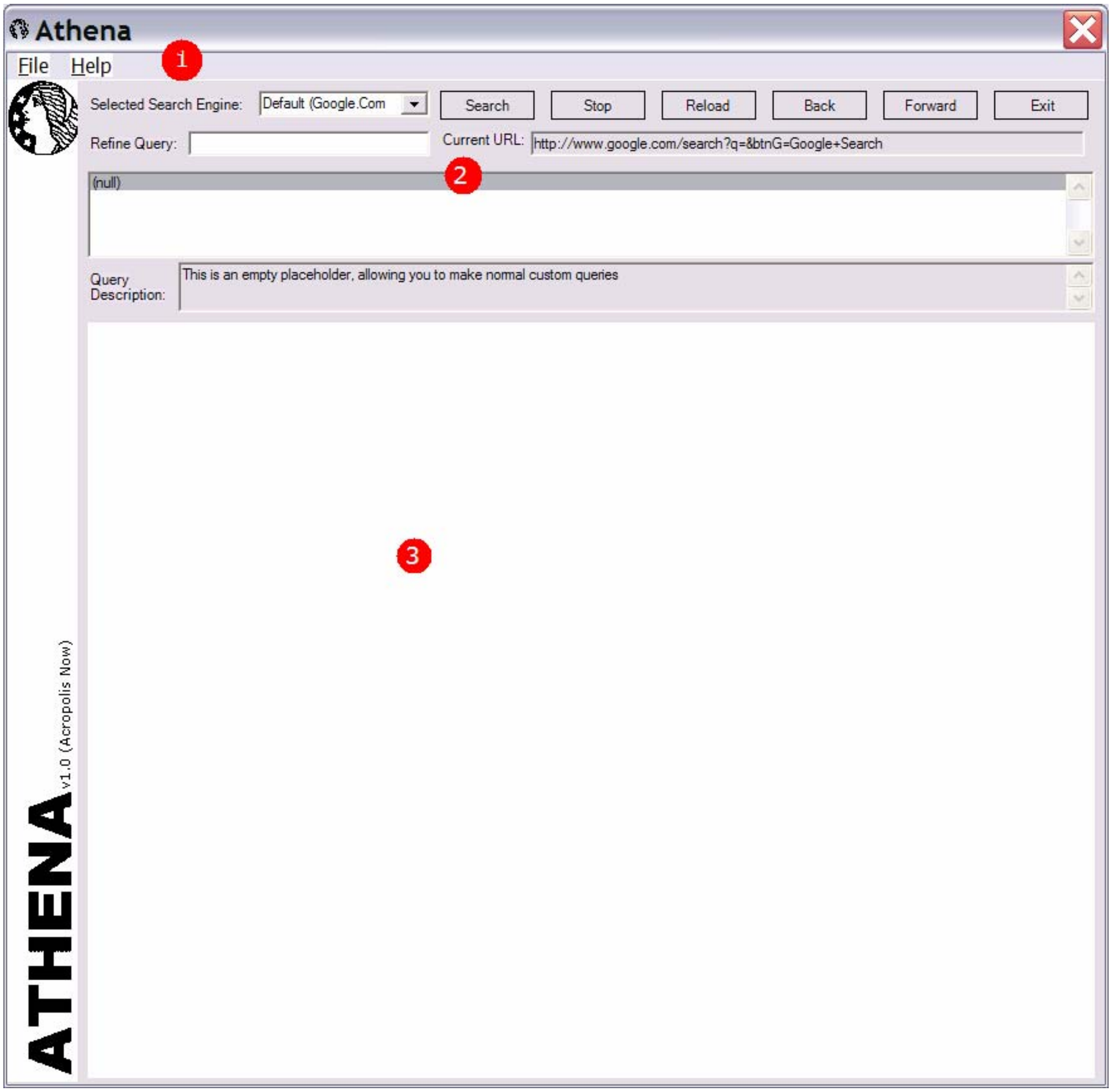
Athena couldn’t have been written without the help of a number of people. Firstly, the guys at Foundstone who’s excellent SiteDigger almost met my requirements and motivated me to write Athena, jonny.ihackstuff.com for the inspiration, same goes for the Fravia guys at searchlore.org. Thanks also go to Muad’dib and Jamaica Dave for being great beta testers and of course to my wife Ozge, who’s put up with my blobbing as I completely fail to juggle this, home and work.

Polite Notice

Athena could be misused to do some really bad things. **Please don’t**. Every time you exploit something through Athena, you give me less of an incentive to publicly release updates. If I get enough complaints from search engine owners, I’ll take it offline.

Getting Started

Installing Athena is easy, just double-click on the installer and follow the on-screen instructions. Once it’s installed, go to Start-Programs-Athena-Athena or double-click the Athena Icon on your desktop. You should get something like the picture below:



Selected Search Engine: Default (Google.Com

Current URL: http://www.google.com/search?q=&btnG=Google+Search

(null)

Query Description: This is an empty placeholder, allowing you to make nomal custom queries

Figure 1: Athena’s startup screen

The screen layout

The screen is divided into 3 areas (shown in Figure 1): The menu(1), query management(2) and the browser view(3). The menu allows you to load xml configuration files, specify output files, exit and see an about page.

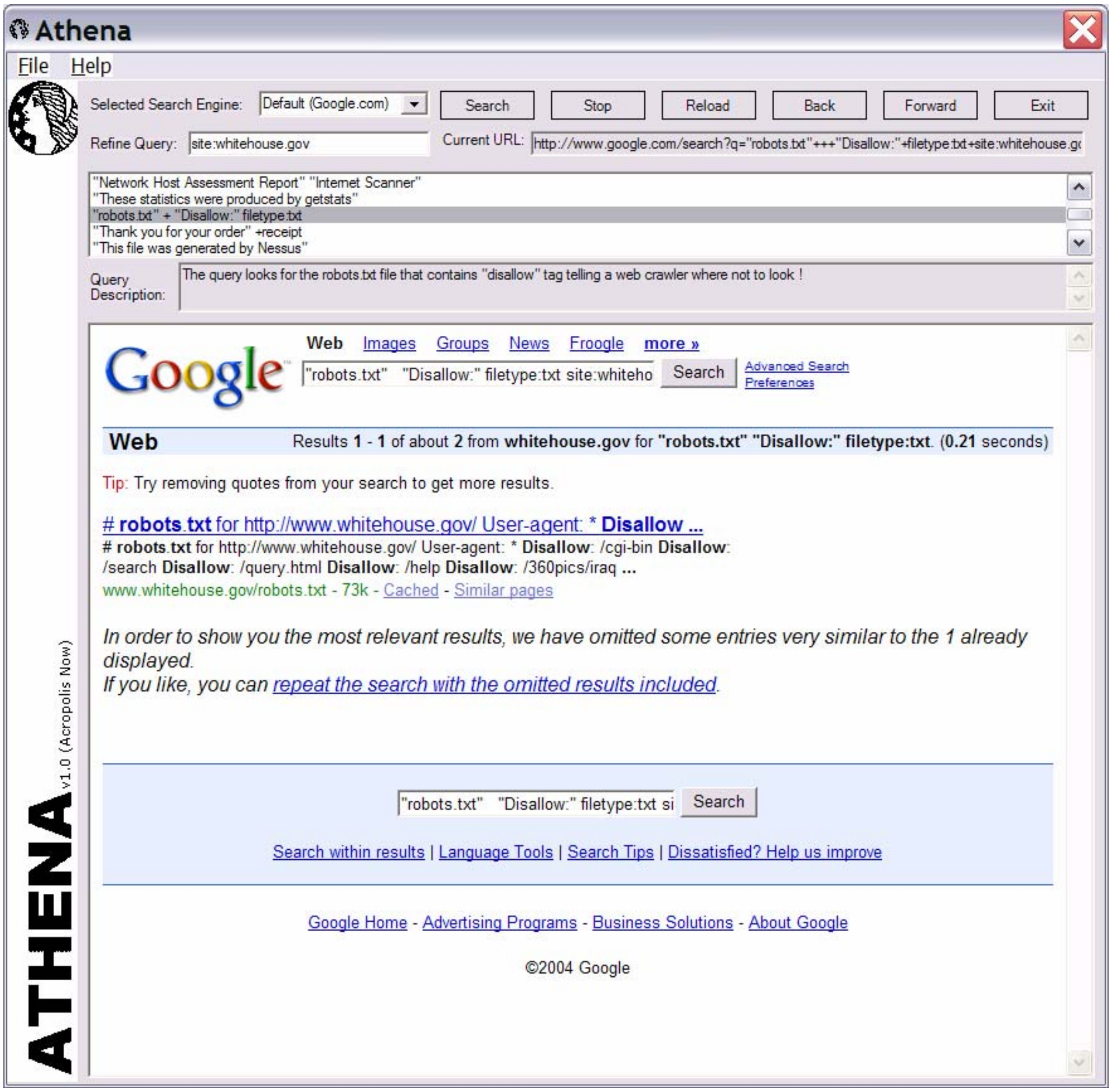
The browser window is effectively an embedded Internet Explorer instance, managed by the values in the query management area.

The query management area is where everything happens. The action buttons (Search, Stop, Reload, Back, Forward and Exit) are fairly self-explanatory. You can select which search engine supported by your configuration file you want to use by using the Selected Search Engine drop-down box. The Query Description Text box provides a description of the refined query selection in the Query Selection box. Whenever a change is made that will affect a URL to be submitted, the Current URL box gets updated.

The default (with no configuration file loaded) settings are to use google.com with no preset query types. This is so you can play around and get a feel for Athena straight away.

Your first query with Athena

Using Athena is easy. Go to File-Open and open google.xml. Athena is now set up to search using google. In the selected search engine drop down box there will be various google sites to choose from. Stick to the Default (Google.com). Scroll down the Query Selection list and select "robots.txt"   "Disallow:" filetype:txt. In the Refine Query box, type site:whitehouse.gov and click on search. The query description states that this query looks for robots.txt files that contain disallow fields, telling the search engine where **not** to look. Our use of site: whitehous.gov restricts the search to the whitehouse.gov domain and is a google specific extension. Figure 2 shows the results.



Selected Search Engine: Default (Google.com)

Refine Query: site:whitehouse.gov

Current URL: http://www.google.com/search?q="robots.txt"+++"Disallow:"+filetype:txt+site:whitehouse.gc

"Network Host Assessment Report" "Internet Scanner"
"These statistics were produced by getstats"
"robots.txt" + "Disallow:" filetype:txt
"Thank you for your order" +receipt
"This file was generated by Nessus"

Query Description: The query looks for the robots.txt file that contains "disallow" tag telling a web crawler where not to look !

"robots.txt"   "Disallow:" filetype:txt site:whiteho

Web Results **1** - **1** of about **2** from **whitehouse.gov** for **"robots.txt" "Disallow:" filetype:txt**. (**0.21** seconds)

Tip: Try removing quotes from your search to get more results.

robots.**txt** for http://www.whitehouse.gov/ User-agent: * **Disallow ...**
robots.**txt** for http://www.whitehouse.gov/ User-agent: * **Disallow**: /cgi-bin **Disallow**: /search **Disallow**: /query.html **Disallow**: /help **Disallow**: /360pics/iraq **...**
www.whitehouse.gov/robots.txt - 73k - Cached - Similar pages

In order to show you the most relevant results, we have omitted some entries very similar to the 1 already displayed.
If you like, you can repeat the search with the omitted results included.

"robots.txt"   "Disallow:" filetype:txt si

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

©2004 Google

Figure 2: Using Athena to identify potential SQL injection

Now select Google (Pages from the UK) from the Selected Search Engine drop-down box. This time there should be no findings. Switch to Google (TR) and Change the refine query entry to site:.tr then hit Search.

Note: Do not click through to any of the sites shown in this tutorial. If you really *must* look, use the google cache.

Open the yahoo.xml config in Athena. Athena will now use Yahoo for searches. From the Selected Search Engine drop down box choose Yahoo.com. Scroll down the query list, select filetype:xls username password email and hit Search. Using the site: prefix it is possible to restrict searches to specific tlds, domains or subdomains.

Your first query with Athena

Logging with Athena is easy. Go to File-Output Log and choose a place to save your logs. From now on any requests made with the search button will be logged, along with the timestamp and a blank line in case you want to write anything in there. Logs are written to until the program is closed or the log file is changed.

Note: Only use of the search button is logged. Clicking through the browser window or using Internet Explorer itself isn't. If you want something to appear in the logs, use the search button!

Hints and Tips

Using Athena is fairly easy, but here are some tips to help get the most out of it.

NEVER click through from a search result to a site without permission. Some of the searches could generate URLs that if accessed could constitute unethical hacking in either your or the target host's country. At the very least you may well be violating the Acceptable Use Policy of the search engine you're using if your terms are too vague. If you're in any doubt that what you're doing is legal, then don't.

If you're authorised, check the search engine cache as well as the actual page. Sometimes there's more to be found in a cached copy than the real thing.

Only searches using the search button are logged – if you hit the next page of search results in the browser window, it isn't logged. That's why there's a blank line after each log entry, so you can put this stuff in along with your notes.

Learn a bit about the syntax of the search engines your using – being able to refine queries that little bit more to specific targets can yield significantly better results than a massive web-wide search.

Play with the XML configs, write your own search query items and search engine prefix/postfix combinations, then send them to me at steve@buyukada.co.uk!

Athena's Configuration Format

Athena uses an XML file (based on SiteDigger's for compatibility purposes) for its configuration. Actually it'll read almost anything you throw at it as long as certain tags are there; it doesn't have to be strictly valid XML. Athena's configuration files have two sections – Search Engines and Search sections. The searchEngineSignature tag surrounds everything. The searchEngine tags are the wrappers for Search Engine definitions. There are only 3 tags used in the searchEngine section: searchEngineName, searchEnginePrefix and searchEnginePostfix. These entries have to be there – the order doesn't matter too much. Consider the following examples:

```
<searchEngine>
        <searchEngineName>Google (UK)</searchEngineName>
        <searchEnginePrefixUrl>http://www.google.co.uk/search?q=</searchEnginePrefixUrl>
        <searchEnginePostfixUrl>%26ie=UTF-8%26hl=en%26meta=</searchEnginePostfixUrl>
</searchEngine>
<searchEngine>
        <searchEngineName>Google (Pages from the UK)</searchEngineName>
        <searchEnginePrefixUrl>http://www.google.co.uk/search?hl=en%26ie=UTF-
8%26q=</searchEnginePrefixUrl>
        <searchEnginePostfixUrl>%26btnG=Search%26meta=cr%3DcountryUK%7CcountryGB</searchEn
ginePostfixUrl>
</searchEngine>
```

The above is fine. As is the above without the <searchEngine> and </searchEngine tags. However, if one searchEngineName follows another then the second searchEngineName is the one added to the drop down box. Hey, I told you it was bad code! Speaking of bad code, one of the issues with using XML is that .NET's XML reader really hates ampersands. I'll fix it eventually, but in the meantime you need to search and replace all & symbols with %26 throughout your XML file, otherwise it'll crash spectacularly when it tries to load the XML in.

The Search section is completely compatible with Foundstone's SiteDigger. This uses the format below. For the sake of readability, the tags used by Athena are shown below. If you create your own XML files please add the rest of the tags. Even if you leave them blank SiteDigger will still read them.

```
<signature>
        <signatureReferenceNumber>23</signatureReferenceNumber>
        <categoryref>T2</categoryref>
        <category>TECHNOLOGY PROFILE</category>
        <querytype>DON</querytype>
        <querystring>intitle:index.of master.passwd</querystring>
        <shortDescription>HTTP Access Password File</shortDescription>
        <textualDescription>This query looked for a directory listing that might contain a
password file.</textualDescription>
        <cveNumber>1000</cveNumber>
        <cveLocation>http://www.1000.com</cveLocation>
</signature>
```

ChangeLog

13/06/04 – Athena 1.0 released. Fixed the following bugs:
Crash when Query List box is selected after loading a fresh config without selecting a search engine from the drop down box first.
Logging prints extra newline for each entry
Got multiple search engine support working (Almost) completely properly

05/06/04 – Athena 0.6. Fixed bugs:
Lotsa crashes fixed. Removed Search engine title and replaced with drop-down list… which doesn't work too well.
Removed GoogleHack references, got Prefix working, but not implemented Postfix yet.

01/06/04 – Athena 0.5.
Multiple Search engines sorted, but one config file needed for Google.com, another for .co.uk, another for .com.tr etc. Works with Yahoo! Put some exception catching in, along with a large number of Duct tape grade kludges.

28/05/04 – Athena 0.1b
Exceptions? We don't need no stinking exceptions! Lots of crashes. OFD cancel causes all containers to clear! Ampersand bug in XML reader.

23/05/04 – GoogleHack 0.2 (Athena 0.1)
Basic browser window, query list box for searches. Queries are hard coded into the program – need a config file.

20/04/04 – GoogleHack 0.1
Uses babelfish and google translator to search google. Proof of concept.

Todo

Code cleanup
More search engine configs (send me yours!)
Implement SiteDigger XML category drop-box to make query fragment finding quicker – but not sure if this is a good idea.