Google Hacking 101

Google Search Basics

Google being a full-text search engine, it indexes entire web pages instead of just titles and descriptions. This allows comprehensive searches based upon key (query) words. Straight from the Google search page.

Google's Boolean default is `AND`; that means if you enter query words without modifiers, Google will search for all of them.

If you search for:
```
snowblower Honda "Green Bay"
```
Google will search for all the words.

If you want to specify that either word is acceptable, you put an `OR` between each item:
```
snowblower OR snowmobile OR "Green Bay"
```

If you want to definitely have one term and have one of two or more other terms, you group them with parentheses, like this:
```
snowblower (snowmobile OR "Green Bay")
```

This query searches for the word "snowmobile" or phrase "Green Bay" along with the word "snowblower."

A stand-in for `OR` borrowed from the computer programming realm is the `|` (pipe) character, as in:
```
snowblower (snowmobile | "Green Bay")
```

If you want to specify that a query item must not appear in your results, use a `-` (minus sign or dash).
```
snowblower snowmobile -"Green Bay"
```
This will search for pages that contain both the words "snowblower" *and* "snowmobile," but not the phrase "Green Bay."

Google Syntax Words

Google also allows keyword searches in specific parts of web pages using special syntax words. Additional commands, called special syntaxes, let Google users search specific parts This comes in handy when you're dealing with billions of web pages and need every opportunity to narrow your search results. Specifying that your query words must appear only in the title or URL of a returned web page is a great way to have your results get very specific without making your keywords themselves too specific.

intitle:

`intitle:` restricts your search to the titles of web pages. The variation, `allintitle:` finds pages wherein all the words specified make up the title of the web page. It's probably best to avoid the `allintitle:` variation, because it doesn't mix well with some of the other syntaxes.

```
intitle:"george bush"
allintitle:"money supply" economics
```

inurl:

`inurl:` restricts your search to the URLs of web pages. This syntax tends to work well for finding search and help pages, because they tend to be rather regular in composition. An `allinurl:` variation finds all the words listed in a URL but doesn't mix well with some other special syntaxes.

```
inurl:help
allinurl:search help
```

intext:

`intext:` searches only body text (i.e., ignores link text, URLs, and titles). There's an `allintext:` variation, but again, this doesn't play well with others. While its uses are limited, it's perfect for finding query words that might be too common in URLs or link titles.

```
intext:"yahoo.com"
intext:html
```

inanchor:

`inanchor:` searches for text in a page's link anchors. A link anchor is the descriptive text of a link.

For example, the link anchor in the HTML code `<a href="http://www.oreilly.com>O'Reilly and Associates</a>` is "O'Reilly and Associates."

```
inanchor:"tom peters"
```

site:

`site:` allows you to narrow your search by either a site or a top-level domain. AltaVista, for example, has two syntaxes for this function (`host:` and `domain:`), but Google has only the one.

```
site:loc.gov
site:thomas.loc.gov
site:edu
site:nc.us
```

link:

`link:` returns a list of pages linking to the specified URL. Enter `link:www.google.com` and you'll be returned a list of pages that link to Google. You can includethe `http://` bit; you don't need it, and, indeed, Google appears to ignore it even if you do put it in. `link:` works just as well with "deep" URLs—http://www.raelity.org/apps/blosxom/ for instance—as with top-level URLs such as *raelity.org*.

cache:

`cache:` finds a copy of the page that Google indexed even if that page is no longer available at its original URL or has since changed its content completely. This is particularly useful for pages that change often.

If Google returns a result that appears to have little to do with your query, you're almost sure to find what you're looking for in the latest cached version of the page at Google.

```
cache:www.yahoo.com
```

daterange:

`daterange:` limits your search to a particular date or range of dates that a page was indexed. It's important to note that the search is not limited to when a page was created, but when it was indexed by Google. So a page created on February 2 and not indexed by Google until April 11 could be found with `daterange:` search on April 11.

Remember also that Google reindexes pages. Whether the date range changes depends on whether the page content changed. For example, Google indexes a page on June 1. Google reindexes the page on August 13, but the page content hasn't changed. The date for the purpose of searching with `daterange:` is still June 1.

Note that `daterange:` works with Julian, not Gregorian dates (the calendar we use every day.) There are Gregorian/Julian converters online, but if you want to search Google without all that nonsense, use the FaganFinder Google interface (http://www.faganfinder.com/engines/google.shtml), offering `daterange:` searching via a Gregorian date pull-down menu. Some of the hacks deal with `daterange:` searching without headaches, so you'll see this popping up again and again in the book.

```
"George Bush" daterange:2452389-2452389
neurosurgery daterange:2452389-2452389
```

filetype:

`filetype:` searches the suffixes or filename extensions. These are usually, but not necessarily, different file types. I like to make this distinction, because searching for `filetype:htm` and `filetype:html` will give you different result counts, even though they're the same file type. You can even search for different page generators, such as ASP, PHP, CGI, and so forth—presuming the site isn't hiding them behind redirection and proxying. Google indexes several different Microsoft formats, including: PowerPoint (PPT), Excel (XLS), and Word (DOC).

```
homeschooling filetype:pdf
"leading economic indicators" filetype:ppt
```

related:

`related:`, as you might expect, finds pages that are related to the specified page. Not all pages are related to other pages. This is a good way to find categories of pages; a search for `related:google.com` would return a variety of search engines, including HotBot, Yahoo!, and Northern Light.

```
related:www.yahoo.com
related:www.cnn.com
```

info:

`info:` provides a page of links to more information about a specified URL. Information includes a link to the URL's cache, a list of pages that link to that URL, pages that are related to that URL, and pages that contain that URL. Note that this information is dependent on whether Google has indexed that URL or not. If Google hasn't indexed that URL, information will obviously be more limited.

```
info:www.oreilly.com
info:www.nytimes.com/technology
```

phonebook:

`phonebook:`, as you might expect, looks up phone numbers.

```
phonebook:John Doe CA
phonebook:(510) 555-1212
```

Google Search Form

<!-- Search Google -->
<form method="get" action="http://www.google.com/search">
<input type="text" name="q" size=31 maxlength=255 value="">
<input type="submit" name="sa" value="Search Google">
</form>
<!-- Search Google -->

Save it as search.htm and double click on it.

A more specific form:

```
<!-- Custom Google Search Form-->
<form method="get" action="http://www.google.com/search">
<input type="text" name="q" size=31 maxlength=255 value="">
<br />
Search for file type:
<select name="as_filetype">
<option value="ppt">PowerPoint</option>
<option value="xls">Excel</option>
<option value="doc">Word</option>
</select>
<br />
Search site:
<select name="as_sitesearch"></option>
<option value="tompeters.com">TomPeters.com</option>
<option value="state.ca.us">State of California</option>
<option value="loc.gov">The Library of Congress</option>
</select>
<input type="hidden" name="num" value="100">
</form>
<!-- Custom Google Search Form-->
```

Google Hacking DataBase (GHDB)

Google Hacking refers to the practice of using search engines, like Google and Bing, in order to discover vulnerable web pages and critical information. It's based on the idea that search engines index a lot of public pages and files, making their discovery a simple matter of building the correct query. Simply place the search string from a database in the Search box and you're on your way.

For example, it's trivial to look for a specific type of file (*filetype:*), on a specific domain (*site:*), with a specific name (*inurl:*), containing a certain string (*intext:*).

The Google Hacking Database (GHDB) was started by Johnny Long, who also published books on the matter, but is now mantained and updated at Exploit Database. The strings are constantly updated. The Google Hacking Database (GHDB) is a compiled list of common mistakes web/server admins make, which can be easily searched by using Google. As a result, you can find things like administrator consoles, password files, credit card numbers, unprotected webcams, etc.

There is also FSDB (Foundstone database). The FSDB is a list of queries that Foundstone has included in addition to the public/commonly known GHDB ones.

GHDB and FSDM contain common search strings for locating vulnerable websites on the Internet, performing DDoS attacks or just general poking around . An example:

https://encrypted.google.com/search?q=filetype:config%20inurl:web.config%20inurl:ftp

Sites: Exploit DataBase and hackersforcharity have more info on the actual queries, how they're structured, and what kind of information you can find. The SiteDigger tool gives an indication as to the type of information you can find, but is not as specific as the above mentioned sites.

See also Google Hacker Tools.