GOOGLE HACKING ETHICS

Google: Book

1 • 2 • 3 • 4 • 5 • 6 • 7 • 8 • 9 • 10 • 11 •

Searching

"passport" filetype:xls site:"*.edu.*" | site:"*.gov.*" | site:"*.com.*" | site:"*.org.*" | site:"*.net.*" | site:"*.mil.*"

"passwordt" filetype:xls site:"*.edu.*" | site:"*.gov.*" | site:"*.com.*" | site:"*.org.*" | site:"*.net.*" | site:"*.mil.*"

Use a search engine to find a page that is describing a relevant database by adding terms likely to appear the page. To find content on income inequality, for example, you might use a query like **“income inequality” (database OR “data set” OR archive OR databank)**. Look for a libguide or other finding tool developed by a librarian or info pro. Since most of these resources include the word guide or libguide in the URL, you can find a useful libguide on, say, entomology, by searching **entomology (inurl:libguides OR inurl:guides OR inurl:researchguides).** In addition, search for mentions of that resource. For example, if you find that FAOSTAT, the statistics portal of the UN Food and Agriculture Organization, is a useful source, try searching for mentions of its URL with a query like **“fao.org/faostat/”** (including the quotation marks).

Factiva is particularly good for finding specially-constructed, complex search queries that their internal search experts have designed for difficult or complex concepts. <more>

Google Has Picked an Answer for You—Too Bad It’s Often Wrong

Going beyond search, the internet giant is promoting a single result over all others, and many are contentious, improbable or laughably incorrect

Certified Ethical Hacker, Google Hacking, hackers for charity.org, Johnny Long ethical hacker, character education, black hat, computer network defender

Google Hacking Mini-Guide By Johnny Long May 7, 2004

Using search engines such as **Google, "search engine hackers"** can easily find exploitable targets and sensitive data. This article outlines some of the techniques used by hackers and discusses how to prevent your site from becoming a victim of this form of information leakage.

Basic Search Techniques

The Google search engine OPERATORS

This article outlines the more harmful applications of the Google search engine, techniques that have collectively been termed "Google hacking." The intent of this article is to educate web administrators and the security community in the hopes of eventually stopping this form of information leakage.

This document is an excerpt of the full *Google Hacker's Guide published by Johnny Long,* was located at http://johnny.ihackstuff.com which has been replaced with http://www.hackersforcharity.org/

VARIOUS OPERATORS AVAILABLE:

- Use the plus sign (+) to force a search for an overly common word. Use the minus sign (-) to exclude a term from a search. *No space follows these signs.*
- To search for a phrase, supply the phrase surrounded by double quotes (" ").
- A period (.) serves as a single-character wildcard.
- An asterisk (*) represents any word—not the completion of a word, as is traditionally used.

Google advanced operators help refine searches. Advanced operators use a syntax such as the following:

operator:*search_term*

Notice that there's no space between the operator, the colon, and the search term.

- The `site:` operator instructs Google to restrict a search to a specific web site or domain. The web site to search must be supplied after the colon.
- The `filetype:` operator instructs Google to search only within the text of a particular type of file. The file type to search must be supplied after the colon. Don't include a period before the file extension.
- The `link:` operator instructs Google to search within hyperlinks for a search term.
- The `cache:` operator displays the version of a web page as it appeared when Google crawled the site. The URL of the site must be supplied after the colon.
- The `intitle:` operator instructs Google to search for a term within the title of a document.
- The `inurl:` operator instructs Google to search only within the URL (web address) of a document. The search term must follow the colon.

Google Hacking Techniques

By using the basic search techniques combined with Google's advanced operators, anyone can perform information-gathering and vulnerability-searching using Google. This technique is commonly referred to as Google hacking.
Site Mapping

To find every web page Google has crawled for a specific site, use the site: operator. Consider the following query:

site:http://www.microsoft.com microsoft

This query searches for the word microsoft, restricting the search to the http://www.microsoft.com web site. How many pages on the Microsoft web server contain the word microsoft? According to Google, all of them! Google searches not only the content of a page, but the title and URL as well. The word microsoft appears in the URL of every page on http://www.microsoft.com. With a single query, an attacker gains a rundown of every web page on a site cached by Google.

There are some exceptions to this rule. If a link on the Microsoft web page points back to the IP address of the Microsoft web server, Google will cache that page as belonging to the IP address, not the http://www.microsoft.com web server. In this special case, an attacker would simply alter the query, replacing the word microsoft with the IP address(es) of the Microsoft web server.

Finding Directory Listings

Directory listings provide a list of files and directories in a browser window instead of the typical text-and graphics mix generally associated with web pages. These pages offer a great environment for deep information gathering (see Figure 1).

Figure 1 A typical directory listing.

Locating directory listings with Google is fairly straightforward. Figure 1 shows that most directory listings begin with the phrase `Index of`, which also shows in the title. An obvious query to find this type of page might be `intitle:index.of`, which may find pages with the term `index of` in the title of the document. Unfortunately, this query will return a large number of false positives, such as pages with the following titles:

- Index of Native American Resources on the Internet
- LibDex—Worldwide index of library catalogues
- Iowa State Entomology Index of Internet Resources

Judging from the titles of these documents, it's obvious that not only are these web pages intentional, they're also not the directory listings we're looking for. Several alternate queries provide more accurate results:

```
intitle:index.of "parent directory"
intitle:index.of name size
```

These queries indeed provide directory listings by not only focusing on `index.of` in the title, but on keywords often found *inside* directory listings, such as `parent directory`, `name`, and `size`. Obviously, this search can be combined with other searches to find files of directories located in directory listings.

Versioning: Obtaining the Web Server Software/Version

The exact version of the web server software running on a server is one piece of information an attacker needs before launching a successful attack against that web server. If an attacker connects directly to that web server, the HTTP (web) headers from that server can provide this essential information. It's possible, however, to retrieve similar information from Google's cache without ever connecting to the target server under investigation. One method involves using the information provided in a directory listing.

Figure 2 shows the bottom line of a typical directory listing. Notice that the directory listing includes the name of the server software as well as the version. An adept web administrator can fake this information, but often it's legitimate, allowing an attacker to determine what attacks may work against the server.

Directory listing `server.at` example.

This example was gathered using the following query:

```
intitle:index.of server.at
```

This query focuses on the term `index of` in the title and `server at` appearing at the bottom of the directory listing. This type of query can also be pointed at a particular web server:

```
intitle:index.of server.at site:aol.com
```

The result of this query indicates that `gprojects.web.aol.com` and `vidup-r1.blue.aol.com` both run Apache web servers.

It's also possible to determine the version of a web server based on default pages installed on that server. When a web server is installed, it generally will ship with a set of default web pages, like the Apache 1.2.6 page shown in Figure 3:

Figure 3 Apache test page.

These pages can make it easy for a site administrator to get a web server running. By providing a simple page to test, the administrator can simply connect to his own web server with a browser to validate that the web server was installed correctly. Some operating systems even come with web server software already installed. In this case, an Internet user may not even realize that a web server is running on his machine. This type of casual behavior on the part of an Internet user will lead an attacker to rightly assume that the web server is not well maintained, and by extension is insecure. By further extension, the attacker can assume that the entire operating system of the server may be vulnerable by virtue of poor maintenance.

The following table provides a brief rundown of some queries that can locate various default pages.

Apache Server Version	**Query**
Apache 1.3.0–1.3.9	`Intitle:Test.Page.for.Apache It.worked! this.web.site!`
Apache 1.3.11–1.3.26	`Intitle:Test.Page.for.Apache seeing.this.instead`
Apache 2.0	`Intitle:Simple.page.for.Apache Apache.Hook.Functions`
Apache SSL/TLS	`Intitle:test.page "Hey, it worked !" "SSL/TLS-aware"`

Many IIS servers	`intitle:welcome.to intitle:internet IIS`
Unknown IIS server	`intitle:"Under construction" "does not currently have"`
IIS 4.0	`intitle:welcome.to.IIS.4.0`
IIS 4.0	`allintitle:Welcome to Windows NT 4.0 Option Pack`
IIS 4.0	`allintitle:Welcome to Internet Information Server`
IIS 5.0	`allintitle:Welcome to Windows 2000 Internet Services`
IIS 6.0	`allintitle:Welcome to Windows XP Server Internet Services`
Many Netscape servers	`allintitle:Netscape Enterprise Server Home Page`
Unknown	`allintitle:Netscape FastTrack Server Home`

Netscape server	`Page`

Using Google as a CGI Scanner

To accomplish its task, a CGI scanner must know what exactly to search for on a web server. Such scanners often utilize a data file filled with vulnerable files and directories like the one shown below:

/cgi-bin/cgiemail/uargg.txt /random_banner/index.cgi /random_banner/index.cgi /cgi-bin/mailview.cgi /cgi-bin/maillist.cgi /cgi-bin/userreg.cgi /iissamples/ISSamples/SQLQHit.asp /iissamples/ISSamples/SQLQHit.asp /SiteServer/admin/findvserver.asp /scripts/cphost.dll /cgi-bin/finger.cgi

Combining a list like this one with a carefully crafted Google search, Google can be used as a CGI scanner. Each line can be broken down and used in either an `index.of` or `inurl` search to find vulnerable targets. For example, a Google search for this:

```
allinurl:/random_banner/index.cgi
```

returns the results shown in Figure 4.

Figure 4 Sample search using a line from a CGI scanner.

A hacker can take sites returned from this Google search, apply a bit of hacker "magic," and eventually get the broken `random_banner` program to cough up any file on that web server, including the password file, as shown in Figure 5.

Figure 5 Password file captured from a vulnerable site found using a Google search.

Note that actual exploitation of a found vulnerability crosses the ethical line, and is not considered mere web searching.

Of the many Google hacking techniques we've looked at, this technique is one of the best candidates for automation, because the CGI scanner vulnerability files can be very large. The gooscan tool, written by j0hnny, performs this and many other functions. Gooscan and automation are discussed below.

Google Automated Scanning

Google frowns on automation: "You may not send automated queries of any sort to Google's system without express permission in advance from Google. Note that 'sending automated queries' includes, among other things:

- using any software which sends queries to Google to determine how a web site or web page 'ranks' on Google for various queries;
- 'meta-searching' Google; and
- performing 'offline' searches on Google."

Any user running an automated Google querying tool (with the exception of tools created with Google's extremely limited API) must obtain express permission in advance to do so. It's unknown what the consequences of ignoring these terms of service are, but it seems best to stay on Google's good side.

Gooscan

Gooscan is a UNIX (Linux/BSD/Mac OS X) tool that automates queries against Google search appliances (which are not governed by the same automation restrictions as their web-based brethren). For the security professional, gooscan serves as a front end for an external server assessment and aids in the information-gathering phase of a vulnerability assessment. For the web server administrator, gooscan helps discover what the web community may already know about a site thanks to Google's search appliance.

For more information about this tool, including the ethical implications of its use, see http://johnny.ihackstuff.com

Google Hacking Mini-Guide Date: May 7, 2004 By Johnny Long.

Googledorks

The term "googledork" was coined by the author and originally meant "An inept or foolish person as revealed by Google." After a great deal of media attention, the term came to describe those who "troll the Internet for confidential goods." Either description is fine, really. What matters is that the term ***googledork*** **conveys the concept that sensitive stuff is on the web, and Google can help you find it.** The official

googledorks page http://johnny.ihackstuff.com/googledorks lists many different examples of unbelievable things that have been dug up through Google by the maintainer of the page, Johnny Long. Each listing shows the Google search required to find the information, along with a description of why the data found on each page is so interesting.

GooPot

The concept of a *honeypot* is very straightforward. According to http://www.techtarget.com, "A honey pot is a computer system on the Internet that is expressly set up to attract and 'trap' people who attempt to penetrate other people's computer systems."

To learn how new attacks might be conducted, the maintainers of a honeypot system monitor, dissect, and catalog each attack, focusing on those attacks that seem unique.

An extension of the classic honeypot system, a web-based honeypot or "page pot" (click here to see what a page pot may look like) is designed to attract those employing the techniques outlined in this article. The concept is fairly straightforward. Consider a simple googledork entry like this:

```
inurl:admin inurl:userlist
```

This entry could easily be replicated with a web-based honeypot by creating an `index.html` page that referenced another `index.html` file in an `/admin/userlist` directory. If a web search engine such as Google was instructed to crawl the top-level `index.html` page, it would eventually find the link pointing to `/admin/userlist/index.html`. This link would satisfy the Google query of `inurl:admin inurl:userlist`, eventually attracting a curious Google hacker.

The referrer variable can be inspected to figure out how a web surfer found a web page through Google. This bit of information is critical to the maintainer of a page pot system, because it outlines the exact method the Google searcher used to locate the page pot system. The information aids in protecting other web sites from similar queries.

GooPot, the Google honeypot system, uses enticements based on the many techniques outlined in the googledorks collection and this document. In addition, the GooPot more closely resembles the juicy targets that Google hackers typically go after. Johnny Long, the administrator of the googledorks list, utilizes the GooPot to discover new search types and to publicize them in the form of googledorks listings, creating a self-sustaining cycle for learning about and protecting from search engine attacks.

Although the GooPot system is currently not publicly available, expect it to be made available early in the second quarter of 2004.

Protecting Yourself from Google Hackers

The following list provides some basic methods for protecting yourself from Google hackers:

- **Keep your sensitive data off the web!** Even if you think you're only putting your data on a web site temporarily, there's a good chance that you'll either forget about it, or that a web crawler might find it. Consider more secure ways of sharing sensitive data, such as SSH/SCP or encrypted email.
- **Googledork!** Use the techniques outlined in this article (and the full Google Hacker's Guide) to check your site for sensitive information or vulnerable files. Use gooscan from http://johnny.ihackstuff.com to scan your site for bad stuff, but *first get advance express permission from Google*!Without advance express permission, Google could come after you for violating their terms of service. The author is currently not aware of the exact implications of such a violation. But why anger the "Goo-Gods"?!
- **TIP**

  Check the official googledorks web site on a regular basis to keep up on the latest tricks and techniques.
- **Consider removing your site from Google's index.** The Google webmasters FAQ provides invaluable information about ways to properly protect and/or expose your site to Google. From that page: **"Please have the webmaster for the page in question contact us with proof that he/she is indeed the webmaster**. This proof must be in the form of a root level page on the site in question, requesting removal from Google. Once we receive the URL that corresponds with this root level page, we will remove the offending page from our index." In some cases, you may want to remove individual pages or snippets from Google's index. This is also a straightforward process that can be accomplished by following the steps outlined at http://www.google.com/remove.html
- **Use a `robots.txt` file.** Web crawlers are supposed to follow the robots exclusion standard. This standard outlines the procedure for "politely requesting" that web crawlers ignore all or part of your web site. I must note that hackers may not have any such scruples, as this file is certainly a suggestion. The major search engine's crawlers honor this file and its contents. For examples and suggestions for using a robots.txt file, see http://www.robotstxt.org.

Thanks to God, my family, Seth, and the googledork community for all the support. Happy Googling!
http://www.hackersforcharity.org
j0hnny http://johnny.ihackstuff.com https://twitter.com/ihackstuff

--

Interview With Himanshu Sharma – One of India's Finest Ethical Hacker нαℭкєя BLOG

RELATED LINKS

SUPPLEMENTAL CONCEPTS

 Hot Site Awards