Global Information Assurance Certification Paper

Google hacking: A tool for the Security Professional

Brian Dzik

GIAC Security Essentials Certification

Practical Version 1.4c

Option 1

Submitted 3/15/05

Table of Contents

Abstract

This paper intends to discuss the Google Hacking phenomenon by examining what Google is, where Google came from and why Google is so important to the digital world. It will delve into the depths of advanced Google searching techniques outlining the basics of Google hacking principles. It will investigate some tools to automate the Google hacking experience. Finally it will address some countermeasures to defend ourselves from Google hacking.

Introduction

In today's world of the electronic marketplace, having an internet presence is crucial to running a successful business. An internet presence is worthless unless it can easily be found, which typically means listing with search engines. One of the most popular and arguably the most powerful of these search engines is Google. Google's appeal is that it returns extremely relevant results, and offers a simple way of generating advanced queries to help users find exactly what they are looking for. These two features have brought the eyes of the internet security world onto Google for use as a security tool.

The amazing results returned by Google, have prompted users to invent new uses for the search engine. Google can be used as a basic port scanner, a game engine, a password generator, and even a cook book. Unfortunately, more sinister uses for Google have evolved. Google has been used by at least one worm writer to intelligently attack vulnerable targets, and is used daily as an application vulnerability scanner.

Johnny Long is one of the foremost experts in the growing field of Google hacking. Johnny and fellow Google hackers gather on one of the original Google hacking websites, ihackGoogle.com. This website is one of the best resources available for search engine hacking. This website claims to be the "center of the Google Hacking Universe", and is responsible for the origination of the term 'googledorks'. "We call them 'googledorks' (gOO gÃ´l'DÃ´rk, noun, slang): An inept or foolish person as revealed by Google. Whatever you call these fools, you've found the center of the Google Hacking Universe!"[1] Googledorks are what makes Google hacking possible. As the definition above explains, these are the individuals that have an insecure web presence indexed by Google.

Google hacking is evidence of a new trend in perimeter testing a network. Testers are going beyond simply testing for vulnerable servers and beginning to extensively test for vulnerable web based applications.

While Google is by no means the only search engine on the internet with the

1 http://johnny.ihackstuff.com/index.php?module=prodreviews

advanced capabilities that make search engine hacking so popular, this paper will focus on Google, and Google hacking. All examples in this paper will be given using the fictional company Foo Widgets, and their website, www.foowidgets.com. For clarity sake, all queries or search strategies will be shown in italics and surrounded by curly braces {}. The braces are not part of the queries, and should be omitted when duplicating these examples. Please note that the terms "hacker" and "hacking" in this paper are used in the classic sense of the words, meaning one who enjoys the intellectual challenge of creatively overcoming or circumventing limitations, and performing that action[2]. These terms do not imply an attacker with malicious intent.

What is Google?

"Google is a play on the word googol, which was coined by Milton Sirotta…. It refers to the number represented by the numeral 1 followed by 100 zeros. Google's use of the term reflects the company's mission to organize the immense, seemingly infinite amount of information available on the web."[3]

Google began its life as a college project called BackRub, written by founders Larry Page and Sergey Brin at Stanford University. While at Stanford, the young Googlers perfected their ideas and developed the basic tenets their multi-billion dollar company would later be built upon. One of these ideas was to use many low-end computers rather than high-end servers to run their search engine.

Initially Larry Page and Sergey Brin had thought to license their technology out to other existing companies. Luckily for them, they didn't find any takers. Yahoo! Founder David Filo suggested they start their own search engine company, so in early September 1998, they did just that. Google.com was launched to great success on September 7th, quickly handling 10,000 queries a day. Less than a year later, partially thanks to some high profile clients, Google.com was handling over 3 million queries a day. In early 2000, Google became the largest search engine in the world, with a billion page index. In November 2004 Google announced it had surpassed another milestone, indexing over 8 billion pages.

Today Google is an international company, employs thousands of people, and is publicly traded on the NASDAQ stock market. Google continues to be a leader in internet technologies, offering such services as Google Groups, an archive of Usenet postings, GMail a searchable web-based email system, and Google Image search, a service that allows the searcher to find images relevant to search terms.

2 http://catb.org/~esr/jargon/html/H/hacker.html

3 http://www.google.com/corporate/history.html

How a Google search works

Google has created a network of linked computers that work together to provide the user with the best search results possible. As shown by figure 1 below taken from Google's technical website[4]; four different types of computers make up this Google Network; web servers, web spiders, index servers, and doc servers. Google's web spiders crawl and cache much of the internet every six to eight weeks, following all of the links they can find. Once this crawl is complete, the crawl is indexed using Google's patented PageRank technology, which determines the importance of each page. On an extremely basic level, PageRank is a voting system. The more pages that link to you, the higher rank you receive. PageRank takes these votes and uses them as part of an advanced proprietary algorithm to determine the actual ranking of each page. When search results are displayed, the higher ranking pages are displayed at the top of the page.

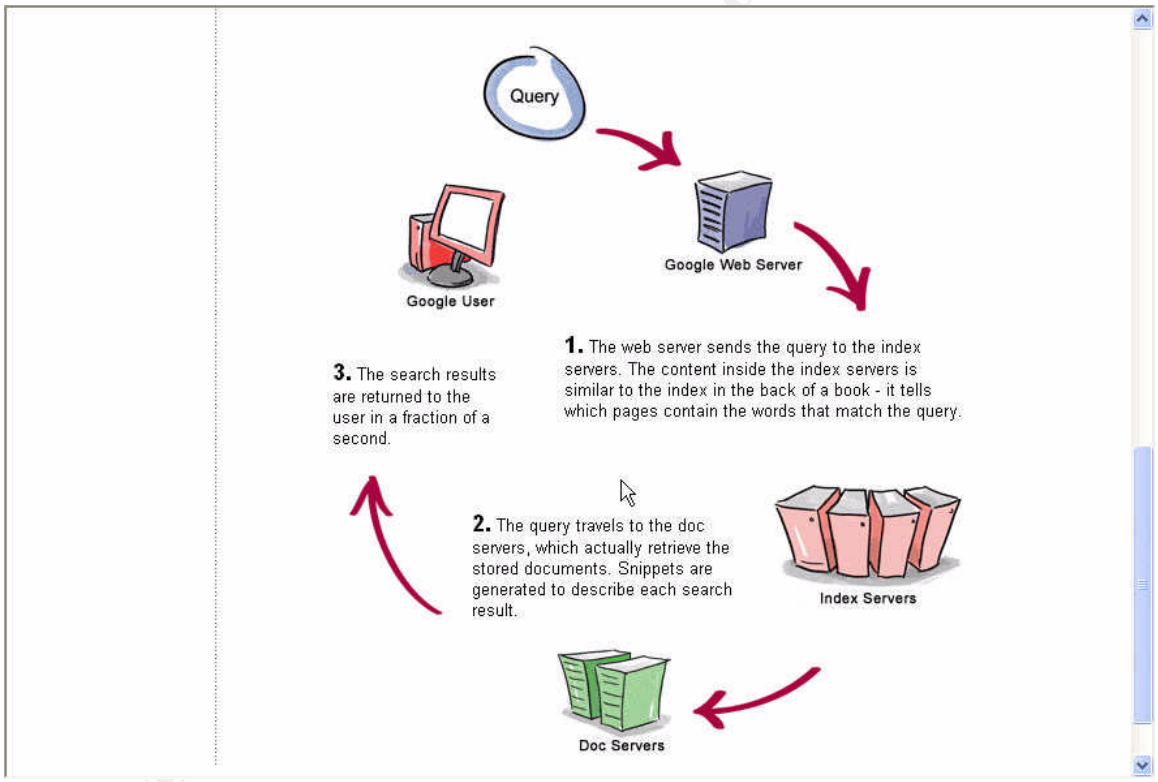


Figure 1 Anatomy of a Google search[5]

When a user types in a search string on the Google search page, that user is interacting with one of the Google web servers. That web server forwards the query to the Google index servers which attempt to match the query to an index of web pages. Once the match has been made, the query is passed onto

4 http://www.google.com/corporate/tech.html

5 http://www.google.com/corporate/tech.html

Google's doc servers. The doc servers are where cached documents are stored, and where the snippet of each web page is created. When the doc servers create the results pages, they are returned to the web server and the user's browser.

Basic Searching

An end user can begin to harness this awesome computing engine with a basic Google search. A basic search is performed by simply entering the search term into the box on Google web site. Google has such confidence in the results that they offer "I'm feeling lucky" button. "Google excels at producing extremely relevant results, and flat out nails many queries such as company names. We're so confident, in fact, that we've installed an "I'm Feeling Lucky" button, which takes you directly to the site of the highest ranked result in your search. Try it and let us know if our confidence is justified."[6] Assuming the user would like to choose the site she visits, Google gives the option to list up to 100 search results per page. Both the "I'm feeling lucky" option and the list option rely heavily on PageRank technology to provide the best results to a search.

Some Googleisms

Search engines default to one of two Boolean defaults: Boolean AND or Boolean OR. Google's default is Boolean AND, which means Google will attempt to match all of your search terms.[7] Another default behavior of Google is that it does not include common words such as 'the' or 'where' in a query. These default behaviors can be modified by using some of the advanced operators and special characters in the next section.

Special characters and Advanced Operators

Basic searching with Google provides impressive results, but Google indexes so many websites, that the results can be a bit overwhelming. Thankfully Google has provided a way to restrict our search results by using some advanced techniques within a query. A word of caution here, Google limits queries to 32 words.

Advanced searching techniques allow a searcher to refine his search to display exactly what we wish to find by using special characters and advanced operators. The first set of these characters, the '+' and '-' characters, include methods for adding or removing specific terms from the query. Do not include a space between these special characters and the term to be modified. The '+' character will return results that include all the search term immediately

6 http://www.google.com/technology/whyuse.html

7 pg 18, Google Pocket Guide; O'Reilly

following the '+'. For example {*Foo +Widgets*} will return all pages that contain both the words Foo and Widgets. The default Google behavior of excluding common words can be overridden with the '+' character. Thus the search string {*Foo +the Widgets*} will include the word "the" in the results. The '-' character omits the term following the '–' in the results. The search {*Foo –Widgets*} will return all pages that contain the term Foo, but do not contain the term widgets.

Google will, by default, attempt to match the terms listed in the search box anywhere on a page to provide results. Including double quotes around the search phrase will force Google to look for an exact match to the terms contained within the quotes. The search {*"foo widgets wobulator"*} will return all pages that contain the exact phrase" foo widgets wobulator", ignoring any punctuation.

The next character represents Boolean *OR* searches. A Boolean OR search can be performed by using either the '|' (pipe) character or the term 'OR'. The Boolean OR search will return results that contain either term surrounding the '|'. *NOTE: the term OR must be used in uppercase*. The search {*Widgets foo OR bar*} will return all pages that contain the word widgets and either the word foo or bar.

Once the searcher has specified an exact query, he may further refine the search by limiting the types of data returned through the use of advanced operators. This paper will only concentrate on the most common operators used in Google hacking. *NOTE: When using the following operators, there is no space between the operator, the colon, and the search term*.

Cache:
As described above, Google caches as many web pages as possible during its crawl of the internet. A searcher can query for results that reside in Google's cache by using the *cache:* operator. Including a search term along with the *cache:* operator forces Google to highlight the search term within the cached page. For instance, {*cache:www.foowidgets.com wobulator*} will display Google's cache of www.foowidgets.com and highlight the word wobulator every place it finds it on the page.

Filetype:
Google will index many different file formats in addition to the standard HTML file. The *filetype:* operator instructs Google to return only the file types specified in the search string. If the searcher does not have the necessary application to read the file format he has chosen, Google will attempt to convert the file to HTML so the file can be displayed in the browser window. The query {*wobulator filetype:xls*} will return all excel spreadsheets in Google's index that include the word wobulator. For more details on the file types Google crawls, visit http://www.google.com/help/faq_filetypes.html.

Link:
During the Indexing process, Google analyzes links between different websites using its PageRank technology. These links are not forgotten once the index is complete. Google will display all of the web pages that link to the target site with the use of the *link:* operator. The query {*link:www.foowidgets.com*} will display all web pages that have a link to the www.foowidgets.com website.

Site:
One of the most useful operators to Google hackers is the *site:* operator. The *site:* operator will limit search results to within a given domain. The search string {*wobulator site:foowidgets.com*} will find any pages that mention wobulator within the foowidgets.com domain.

Intitle:
When looking for a specific type of server, the *intitle:* operator is invaluable. The *intitle:* operator will return pages that include the first term after the operator in the title. Any further terms in the query will not be limited to the title. The query {*intitle:wobulator foo*} will return any documents that have the word wobulator in the title of the page and the word foo anywhere on the page.

Allintitle:
The *allintitle:* operator is very similar to the *intitle:* operator, except that *allintitle:* will only return results that include all terms in the title. The query {*allintitle: wobulator foo bar widgets}* will return pages that have all search terms in the title of the page.

Inurl:
The *inurl:* operator will restrict the results to pages containing that word in the url. For instance, {*inurl:widgets wobulator*} will return pages that mention the word "widgets" in their url, and mention the word "wobulator" anywhere in the document

Allinurl:
The *allinurl:* operator returns is very similar to the *inurl:* operator, except *allinurl:* will return only results that contain all terms in the url. For example, the search string {*allinurl: wobulator foo bar widgets*} will return www.foowidgets.com/wobulator/bar/index.html but not www.foowidgets.com/test/bar/index.html.

Before continuing, one special character needs to be revisited. The '-' character can be added to many advanced operators for a negative affect. For example, a query built with the *–site:* operator will exclude the specified site from the search results.

Google Hacking in its most basic form is the application of advanced searching techniques using the Google search engine. Here are five different types of Google hacking.

Searching for violations of company policy

Most companies today have some sort of company policy that prohibits employees from using the internet for personal use on company time. Some companies also prohibit users from using their work email address to post messages on the internet. The Google hacking techniques outlined in this paper can help determine compliance with these types of policies. A first step would be to simply search for your company's email address on Google's web search. The search strategy *{@foowidgets.com}* will return all pages where an employee has used her @foowidgets.com email address on the internet. This may also return many results from your company site, but we can filter out those results by using the '-' character and the site: operator together. Our search strategy is now {*@foowidgets.com–site:foowidgets.com*}. Once we have exhausted the results from the regular Google search, we can turn our attention to Google Groups. Using the same search strategy,{*@foowidgets.com –site:foowidgets.com*} we can now look for our email addresses posted in newsgroups.

Another search strategy that may flush out policy violations is {*site:foowidgets.com –site:www.foowidgets.com*}. This search strategy excludes your main internet site, while still looking at your domain for rogue web servers. This strategy will find a web server at test.foowidgets.com if it is listed on Google.

These are not the only search strategies that will find policy violations; rather they are generic ideas to start your thought process. Each company should read through its security policy line by line to determine if there are any items that can be revealed by Google, and formulate a specific search strategy for each item.

Brand Integrity and Copyright Infringement

Many companies are very conscious of the image they portray. These organizations benefit from this type of Google hacking. To maintain brand integrity, the first search strategy is simply the company name, excluding the company website *{foowidgets –site:foowidgets.com}*. Expanding on this idea, the search strategy can be refined to include products offered by our company, the wobulator widget in our example. Our search strategy is now *{foowidgets OR wobulator widget –site:foowidgets.com}*. After exhausting the Google web search, we can turn our attention to the Google Image search. The Google Image index analyzes each web page looking for captions of pictures. These captions as well as the image title become our search results. This type of

image search can identify sites that are using our brand in a negative way. These search strategies can also identify copyright infringement, with some expert analysis. The Google images search strategy *{wobulator –site:foowidgets.com}* can return results that look surprisingly like our wobulator product, on a competitors website.

Zero knowledge assessment as part of a Penetration Test

One of the first steps take during a Zero Knowledge assessment is to use Google with the site: operator. Our first search strategy will be {*site:foowidgets.com}*. With this search we will look for employee information such as names, email addresses or phone numbers. We will also look for any hints to the structure of the internal network such as intranet links, or style sheets. The next step is to attempt to get directory listings. Our search strategy for this step is {*site:foowidgets.com intitle:index.of OR intitle:"index of"*}. If we get any results here, we can browse through the directory listings and hopefully see something we aren't supposed to see. If the results are overwhelming, we can further refine our search to look for any commonly used directories. Some of these common directories are /backup, /private, /admin, /administrator, /old, /upload, /download, /secret, and /private. Our search strategy changes to add in the name of the directory we are looking for {*site:foowidgets.com intitle:index.of.private OR intitle:"index of private"*}. It is interesting to note that these steps can all be performed using Google's cache of our target site if available. We can get all of this information without ever having to visit the target's web server. However, as Johnny Long points out in a recent presentation at Shmoocon[8], we must use the correct syntax to remain anonymous. As you can clearly see from figure 2 below, Google is stating that the cache may reference pictures that exist on the original site.

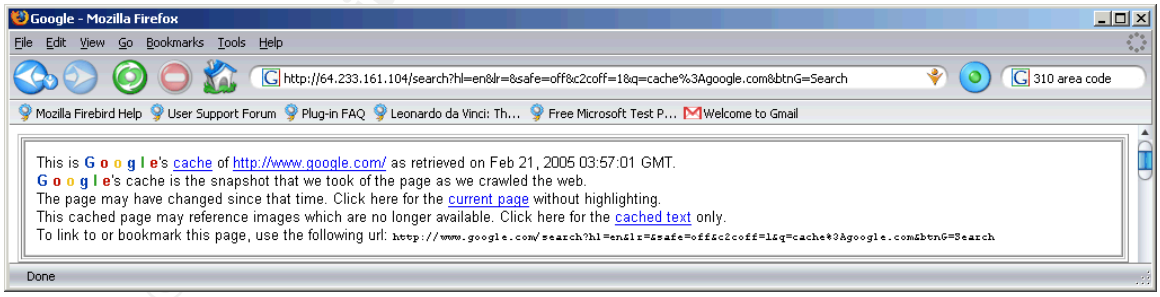


Figure 2 Searching Google's cache

In Johnny's presentation he proves that the images are being displayed from the target website by using a sniffer while visiting the Google cache of a web page. Johnny has discovered a method of viewing the cache of a web page while only visiting Google. The first step is to run a query on the target website without using the site: operator {*www.foowidgets.com*}. This should return a page that

8 http://johnny.ihackstuff.com/security/premium/2005_Shmoocon.pdf

lists all the information that Google has on the target site. Included in this information should be a link that is Google's cache of the website. Next we right click and copy the cached link. Paste the link into the toolbar and append '&strip=1'[9] to the end of the link. '&strip=1' forces Google into displaying the cache in text only mode. Our complete URL is now {http://64.233.161.104/search?q=cache:zhdf23cd34t5:www.foowidgets.com/+&hl=en&strip=1}. We can visit this URL and not touch the target web server at all.

Once the target site of the assessment has been exhausted of information, it is time to turn our focus to searching for any email addresses used on the internet. Using the same search strategy listed above {*@foowidgets.com –site:foowidgets.com*}, we look for any emails, or postings that may help get a foot in the door. Again as above, use the same search strategy to look through Google Groups for any Usenet postings. At some point in the Zero Knowledge assessment, it is important to lookup the domain name registration information. While looking at the registration, note the email address listed in the contact information. The assessment may be on foowidgets.com, but they could have used a different domain name when they opened for business, foo.com for example. If this address is different from what was expected by looking at the domain, we need to use this email domain to search on as well. Our search strategy will change to { *@foowidgets.com OR @foo.com –site:foowidgets.com*}

Google Vulnerability scanning, without a specific target

This concept is what has popularized Google hacking, and it is all thanks to the googledorks. The Google hacker starts building his search strategy upon what he knows about the vulnerable product. He refines his strategy to exclude false positive results from different versions of the software. Next he removes any references to the vulnerability write-ups from security companies that have found the vulnerability and posted it. Finally he removes military sites from his search. For this example, we will be looking for the vulnerable foowidgets email server. Our search strategy will include all sites running the FooWidgets mail server, but exclude any mentions to the CVS write up. We know the vulnerable mail server has the text "FooWidgets Web Portal" in the title of the login page, and the text "Select your language to begin" in the text of the page. {*intitle:"FooWidgets Web Portal" intext:"Select your language to begin." –cvs -site:.mil*}. Google has provided the targets, now its up to the malicious user to take advantage of it.

As an example of a real world scan, consider the search strategy {*ext:pl inurl:cgi intitle:"FormMail *" -"*Referrer" -"* Denied" -sourceforge -error -cvs –input*}[10]. This search string shows the maturity of Google hacking in that it identifies targets, and attempts to exclude any vulnerability alerts and warnings.

9 http://johnny.ihackstuff.com/security/premium/2005_Shmoocon.pdf

10 Google hacking database entry 467

Malicious intent, with a specific target

Chances are, someone wants to break into YOUR network and is looking for an entrance point. This Google hacking type is very similar to the other examples given here, and uses many of the same techniques. Where this example differs, is in the fact that someone on the internet is specifically trying to do harm to you, your company, or your company's systems. There are many different search strategies one can use in this case; the first one simply uses the site: operator. This strategy will return all pages from your site that Google knows about. Most companies have hundreds or even thousands of pages indexed by Google. As pouring through these results can be a bit overwhelming, we will refine our search strategy with some of the advanced operators listed above. For the first refinement, we will be looking at the FooWidgets site for any Excel spreadsheets with the search strategy {*site:foowidgets.com filetype:xls*}. We can re-use the 'filetype:' refinement with any one of the 12 filetypes Google indexes.

Our next refinement starts to look at specific vulnerabilities within our site. Its also a good idea to browse for any business sites that link to the target. A business partner of the target may have a special login that it passes, or other information that can help to get a foot in the door. The search strategy for linked sites is {*link:www.foowidgets.com –site:foowidgets.com*}. Remember to exclude your target site as the results will include links from the target site to itself.

Google Hacking yourself

Of course it is a good practice to use all of the search strategies listed in this section to Google hack yourself, the malicious users certainly will. Wouldn't it be easier for us if there were a tool that could automate many of these processes for us?

Automated tools

Not every company can afford the time and effort it takes to manually Google hack their own company. Many tools and applications are now being written to automate the Google hacking process. The foundation that many of these applications are written on are the Google API's. These API's allow developers to create their own interface to the Google searching system. You must have a Google account to get a license key for the API's, and the terms of service state that your program must transmit your license key with all requests. Any program written using the Google API is limited to 1,000 automated queries a day.

THE most important and powerful tool written for Google hackers is the Google Hacking Database or GHDB. "PASSWORDS, for the LOVE OF GOD!!! Google found PASSWORDS!" [11]The GHDB is the product of contributions by the Google hacking community at ihackgoogle.com. Site creator and one of the original

Google hackers, Johnny Long, and crew work tirelessly to produce the most complete collection of Google hacks available. As a result of this work, the GHDB contains many searches for newly released vulnerabilities, as well as default product configurations such as web cam software. The Google hacking database is available for download as an XML file, allowing other tool authors to concentrate on writing the application, without being a Google hacking expert.

Foundstone Sitedigger

The first and most basic tool is sitedigger, written by Foundstone. Sitedigger is integrated with the Google hacking database and uses the Google API. You will need to provide your Google license key to use sitedigger. The least configurable tool mentioned in this paper, sitedigger only allows you to select a site to test and choose which Google hacking signatures to run against it.

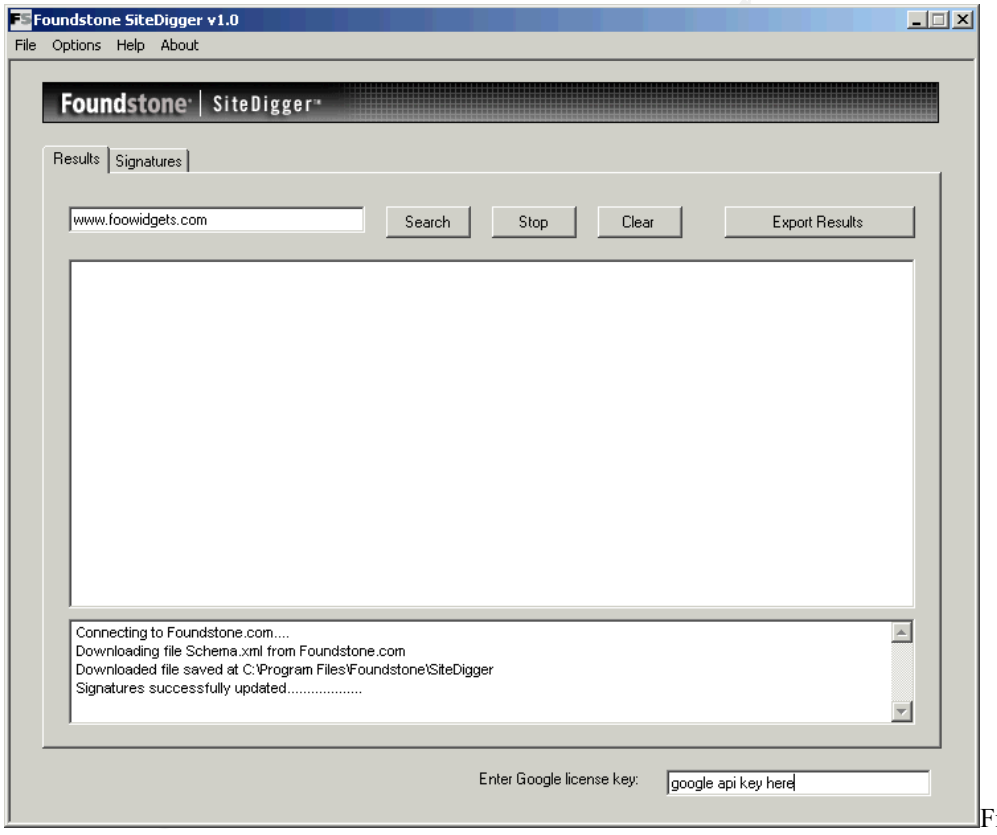


Figure 3
Foundstone SiteDigger

Sitedigger has a built-in feature that update the Google hacking database, and features the best reporting capabilities of the tools reviewed here.

Apollo2.0

11 files containing passwords

Apollo2.0 has one advantage over sitedigger, in that it doesn't need a Google api key to run. This means that the Google hacker can run as many queries as she likes, without worrying about breaking the Google API usage agreements. Apollo is configured by using simple text files, and reporting is limited to simple text format.

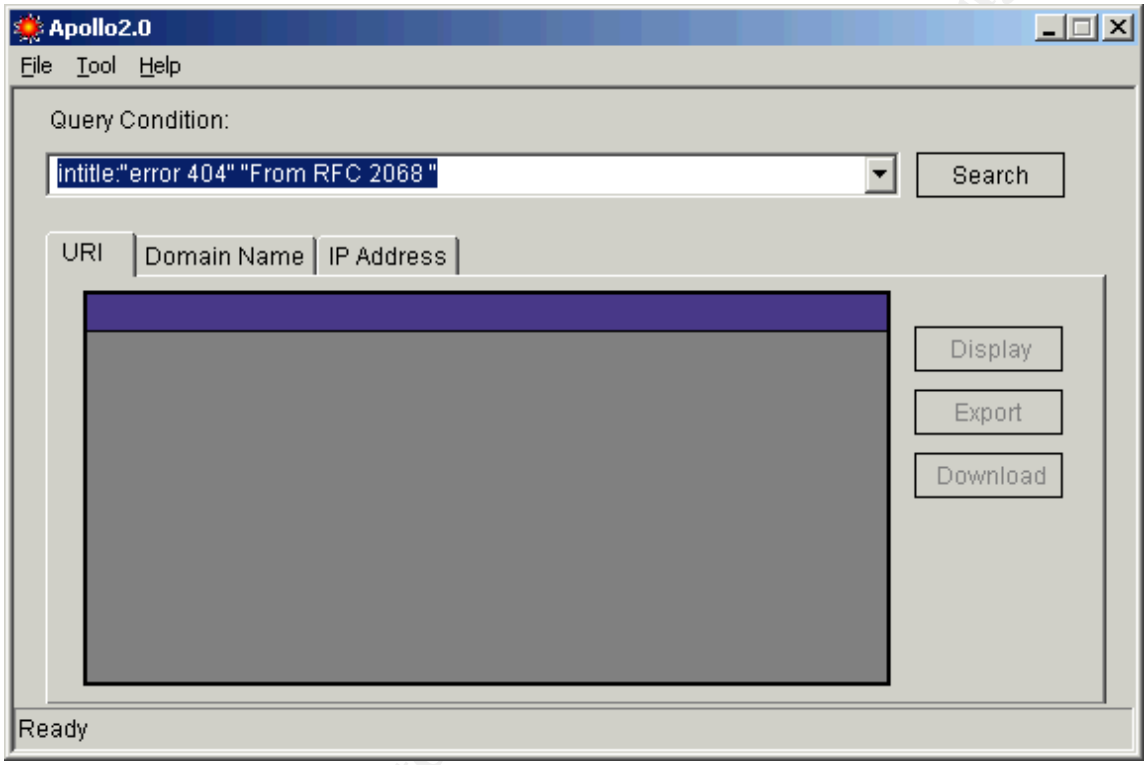


Figure 4 Apollo 2.0

Athena

Snake Oil Labs is another pioneer in the Google hacking space. They have used their vast knowledge of Google hacking to create Athena 2.0 or A2. One advantage to A2 is that it doesn't use the Google API. The A2 download includes an excellent tutorial on how to use the tool, as well as some basic Google hacking techniques. A2 gives you the ability to use the Google Hacking Database for your queries, and lets you easily refine your searches further. For each entry from the GHDB, the authors have given some interesting hints and tidbits, such as default product usernames and passwords to try. There is a notes tab that keeps track of the queries you have run, and lets you set a flag for "interesting" queries. A2 is configured using XML files. Any customizations made to entries from the Google hacking database can be saved for your next session. Athena 2 will store the raw XML that is generated by each search you run thru it so you can repeat the exact search later. The only real drawback to A2 is that it has a built in web browser and cannot be configured to use an

external web browser.

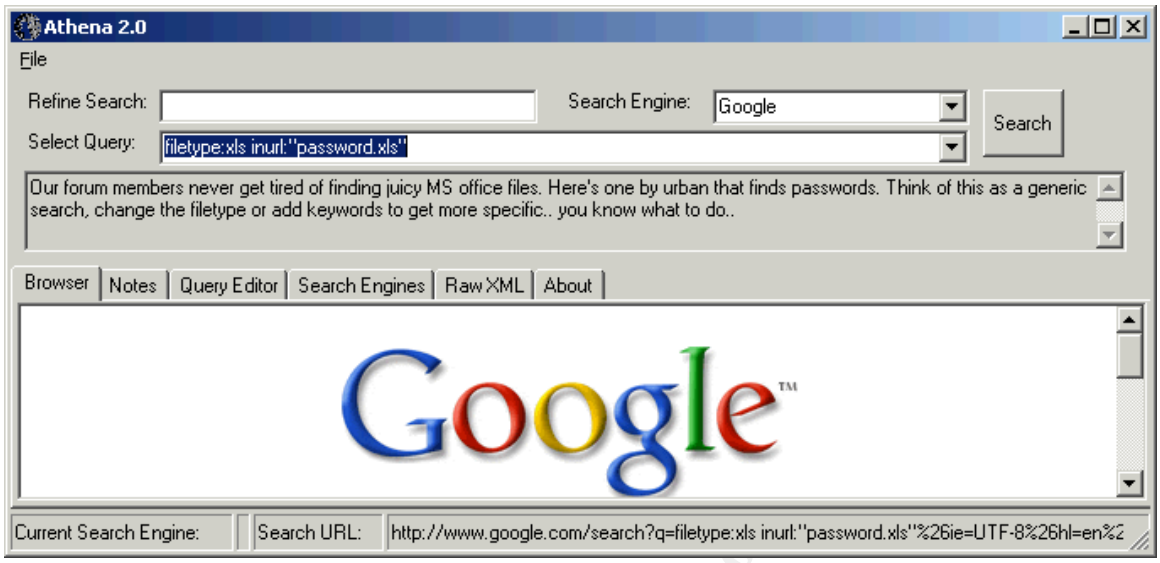


Figure 5 Athena 2.0

Wikto

Wikto is more than just an automated Google hacking tool; it is a complete web assessment tool. WARNING, this tool goes beyond Google hacking, and actually will test the server and any applications running on the server. Use this tool ONLY on servers you have permission to test, or on sites you own. You start using Wikto with an applet called Googler. Googler will search for certain filetypes in the Google index. These filetypes and directories are then imported into an applet called BackEnd. BackEnd attempts to mine a site for common files and directories. The next applet is the Wikto applet, which attempts to emulate a well known Cgi scanner called Nikto. The Wikto applet can be populated by either the Googler or the BackEnd applet. Finally Wikto has an applet called GoogleHacks. GoogleHacks will import the GHDB, and will automatically run all queries from the GHDB on a site of your choice. You can also run manual queries, or choose certain queries from the GHDB. Your search results are displayed in a window at the bottom of the screen.

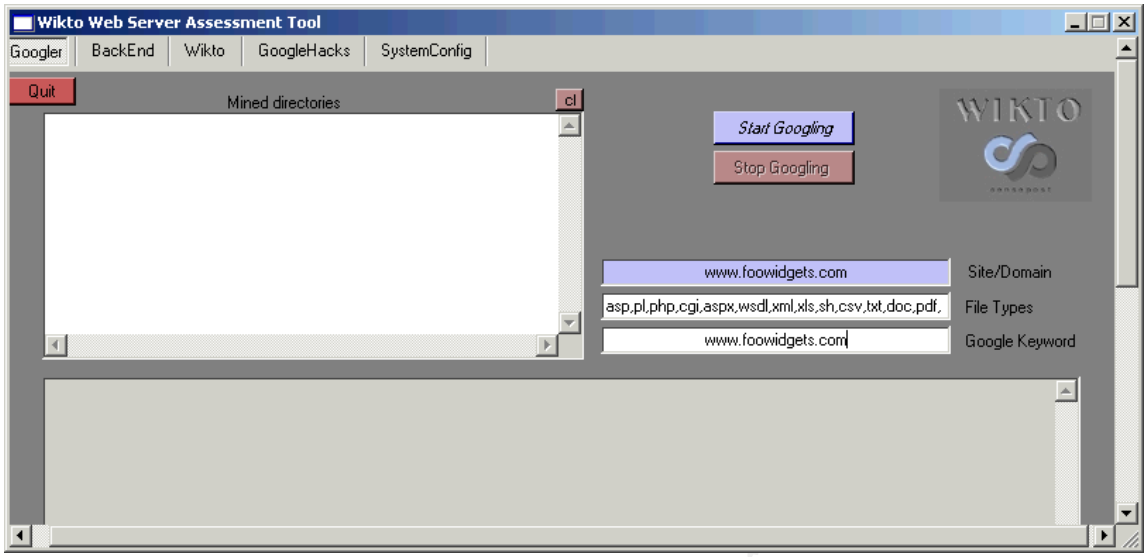
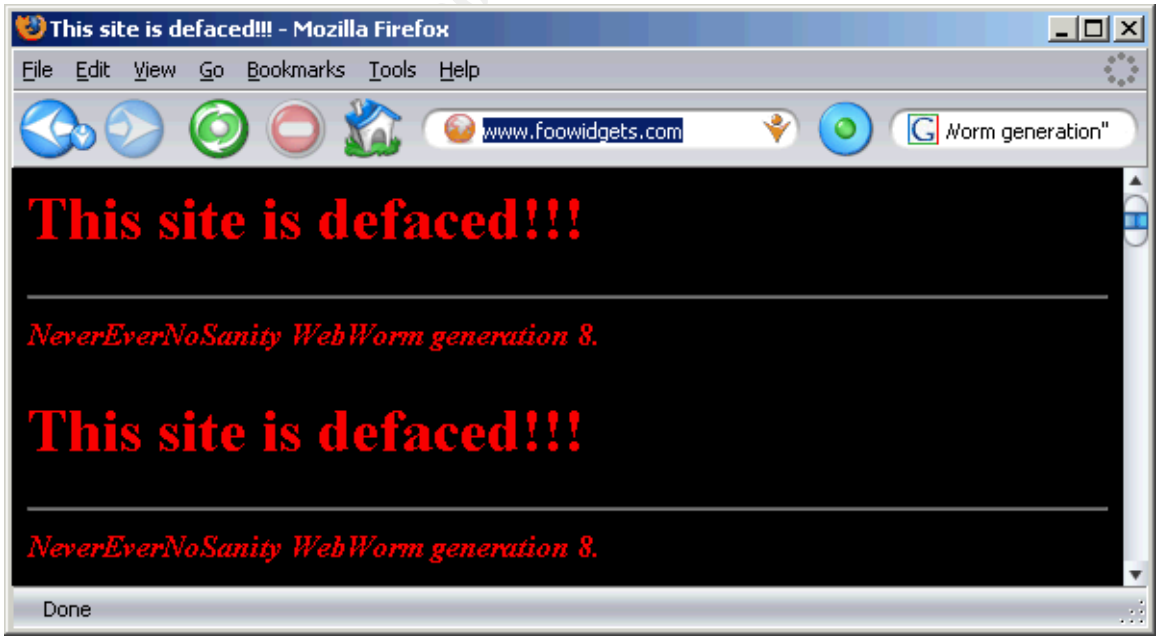


Figure 6 Wikto

The next step in Google hacking; the automated worm

In December 2004 someone in the computer underground took automatic Google hacking techniques to the next level: a worm that can intelligently attack vulnerable targets. "The worm is the latest twist on using Google as an attack tool, a practice known as Google hacking." [12] The Santy.A worm uses the Google search engine to search for viewtopic.php, a common page used by the PHPbb software[13].



12 http://www.zdnet.com.au/news/security/0,2000061744,39175010,00.htm

13 http://www.computerworld.com/securitytopics/security/story/0,10801,98453,00.html

Figure 7 Website defaced by Santy.A worm

When Santy finds a vulnerable website, it defaces it as seen in the figure above. In the text of the defacement, the Santy authors included a “generation” variable. In the above picture, the generation 8 means that this site was compromised by the eighth iteration of a particular thread of the worm. The compromised web server then starts the process all over again by running a specific search string. Many variants of Santy quickly started searching using other search engines and other search strategies.
At the request of many Security companies, Google disabled the search used by Santy and all Santy variants. Google has since taken a more proactive response to automated Google hacking and certain queries are blocked from running.

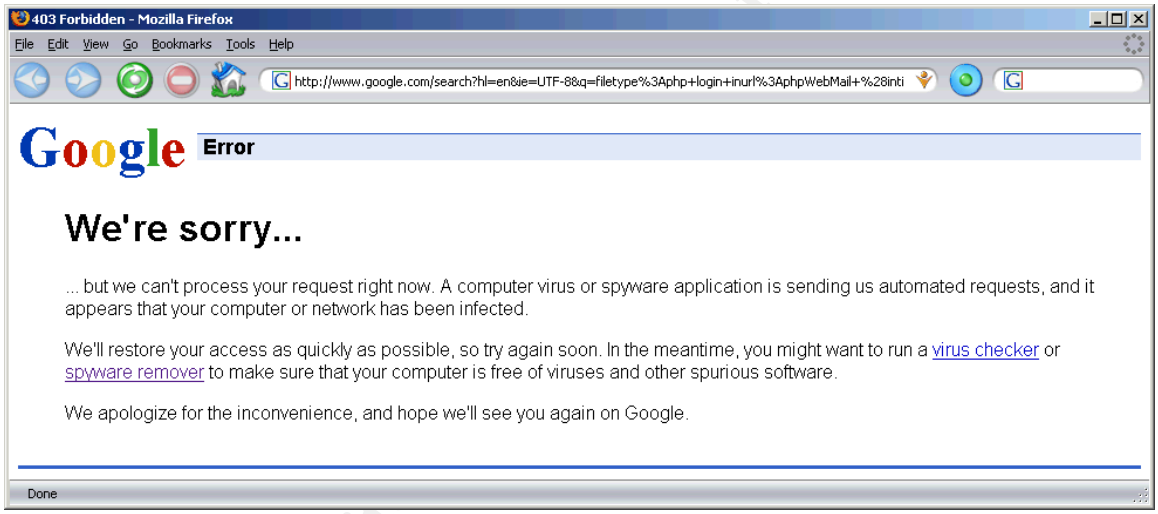


Figure 8 Blocked Google Search

Santy is the first worm to take advantage of Google hacking techniques to find targets, but by no means will it be the last.

Countermeasures

How to defend against Google hacking

After discussing the techniques and the community surrounding Google hacking, we know what information a Google hacker can find on our site, without ever visiting our site. Some important question must be asked. What can be done to protect our web presence against malicious Google hacking? Does Google belong in defense in depth?
There is no “cookie cutter” approach to defend against Google hacking. This paper will identify areas to consider when building a defense, but as each

situation is unique, each company is unique.

Defense in-Depth

We need to include a Google hacking phase in our practice of Defense in-Depth.

One obvious opportunity for inclusion in Defense in-Depth occurs after the installation phases but before and during the implementation, operation and maintenance phases of internet application development[14]. We need to Google hack our application once it has been implemented to determine any new exposures. Once we have assured ourselves that our application will not expose any sensitive information, we can continue on with our implementation plan. In addition, we must add Google hacking search strategies for the new application into our security scanning process for the life cycle of the application.

The Defense in-Depth book of the Security essentials class includes several tips for maintaining Web application security[15]. A Google hacking search strategy can be created to search for each one of these tips.

Company security policy

The first step in protecting your company from Google hacking is a well written Enterprise Security Policy. This policy must include sections prohibiting using company email addresses to post on the internet or on usenet. Each employee must be aware of the consequences of violating the security policy. The security policy should restrict use of the email tool to business purposes. *"E-Mail Privacy*… management may wish to monitor the employee's e-mail for various reasons (e.g., to be sure that it is used for business purposes only or if they are suspected of distributing viruses, sending offensive e-mail, or disclosing organizational secrets.)"[16]

The security policy should state what is allowed to be posted on the internet. As The Defense in Depth book of the Security Essentials course explains, "There is little difference between two competing organizations…what often makes the difference between them is the proprietary information, such as trade secrets and process differences"[17]. These trade secrets must be protected by the company security policy. Once protected in this manner, the area responsible for policy compliance should create specific search strategies, using the techniques outlined in this paper.

Building your defense

---

14 SANS Institute. Track 1 – SANS Security Essentials. Volume 1.2

15 SANS Institute. Track 1 – SANS Security Essentials. Volume 1.2

16 http://csrc.nist.gov/publications/nistpubs/800-12/800-12-html/chapter5.html

17 SANS Institute. Track 1 – SANS Security Essentials. Volume 1.2

Creating policies to dissuade users from disclosing sensitive information is a great first step, but to properly defend a network against Google hackers you must defend against Google itself. You can stop Google, and other search engines that work on the robot principle from scanning your site by putting a file called robots.txt in the root directory of your website. You can specify in robots.txt the parts of your site you don't want Google to crawl.
The robots.txt file must contain a series of records separated by blank lines. The first line of each record is the User-agent, which is the name of the robot the particular record is describing access for[18]. The names of the robots can be found at one of the following websites;
http://www.siteware.ch/webresources/useragents/spiders/ or
http://www.psychedelix.com/agents.html .
The robots.txt file supports wildcards, so it is possible to determine access for all robots by using the "*" in the user-agent field.
The next line of the robots.txt file is the Disallow field. In this field we describe which directories or files we don't want scanned by the search engine's robots (also known as spiders).
Let's step through a few sample robots.txt files.
The following allows all robots to visit all files.

```
User-agent: *
Disallow:
```

This entry will keep all robots out of all directories.

```
User-agent: *
Disallow: /
```

We can specify specific directories that we don't want scanned. The following example, will keep all robots out of the /cgi-bin/ directory, as well as any subdirectories. By including the trailing /, we limit the exclusion to directories.

```
User-agent: *
Disallow: /cgi-bin/
```

By not including the trailing /, we can prevent spiders from crawling files as well. This example will prevent all spiders from crawling the /images directory, as well as any files in the root directory that start with the string images, including images.html and imageshelp.asp.

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /images
```

The following example will stop Google's robots (googlebot) from crawling anything on our site, but allows all other robots access to the whole site.

```
User-agent: googlebot
Disallow: /
```

This final example prevents Google's robots from crawling the wobulator.htm file, but allows all other robots full access to our entire site.

```
User-agent: googlebot
Disallow: wobulator.htm
```

18 http://www.robotstxt.org/wc/norobots.html

Once the robots.txt file is complete, it can be checked for errors on the Robots.txt Validator located at http://www.searchengineworld.com/cgi-bin/robotcheck.cgi.

Building your robots.txt file

There are a few different schools of thought regarding the robots.txt file and security. One way of thinking is to create a directory structure that places all of the directories to be excluded under one directory in the root. Here is a sample

```
User-agent: *
Disallow: /norobots/
```

We can put anything we want in the /norobots/ directory, and it will not be scanned by compliant robots. For example, the www.foowidgets.com/norobots/cgi-bin directory will not be scanned, and there is no mention of it in robots.txt.

"However, in practice this is a bad idea -- it's too fragile. Someone may publish a link to your files on their site… Or someone may misconfigure your server at some future date, "fixing" it to show a directory listing. Which leads me to the real answer:

The real answer is that /robots.txt is not intended for access control, so don't try to use it as such."[19]

Another method of designing a robots.txt file is to include all directories to be excluded in robots.txt, and count on your site security to protect your files. A sample of this type of robots.txt follows.

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /images/
Disallow: /private/
Disallow: /users
Disallow: /webmail/
```

A consideration to this method is that you are basically giving a map to your sensitive directories to anyone who cares to look for it. It is imperative, if you use this approach, to follow secure coding practices, and implement secure code reviews. You must take care to appropriately secure your applications and file systems.

Other removal tricks

19 http://www.robotstxt.org/wc/faq.html#log

Another way to prevent search engines from indexing parts of your site is to use Meta tags within the pages themselves. These tags are placed in the <HEAD> section of an individual web page. One benefit to the Meta tag method is that it does not require assistance from the server administrator. A word of caution however, not all robots will follow these meta tag rules. Meta tags behave similar to the robots.txt method, but offer some advantages as we will see. The first element of the meta tag is the NAME field. This is the area where you name the robot we are describing access for. The META NAME = "ROBOTS" is a wildcard meaning all compliant spiders.
The CONTENT field is where you describe the type of access given to the robots named in the NAME field. The CONTENT="NOINDEX,NOFOLLOW" is the default deny all statement. The following meta tag will prevent all robots from scanning any links on your site.

<META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">

You can allow or deny certain spiders using the meta tag method. The following code will prevent the Google robot from scanning the site.

<META NAME="GOOGLEBOT" CONTENT="NOINDEX, NOFOLLOW">

The meta tag standards document is available at http://www.robotstxt.org/wc/exclusion.html#meta. This document describes in great detail all of the options available for use with meta tags.

Google's robots are compliant with the meta tag standards, and since this paper focuses on Google and Google hacking, we will look at some Google specific tricks that can be performed with meta tags. The first trick is a meta tag that will prevent Google from displaying snippets for our pages. Using this trick will also prevent Google from caching our site but does still allow Google to scan it.

<META NAME="GOOGLEBOT" CONTENT="NOSNIPPET">

The next trick is to allow Google to display snippets, but not allow caching of our site. This tag will still allow Google to scan our site.

<META NAME="GOOGLEBOT" CONTENT="NOARCHIVE">

Secure coding practices

Once we have build our indexing rules using either the robots.txt method or the meta tag method, we must look at the applications running in our web space. This paper does not attempt to cover application security, but will cover some basic best practices. For more detailed papers on application security, please see the SANS reading room.
A common search strategy Google Hackers use is a search for product default

messages, such as the “powered by Apache” message that is included in Apache web server installations. It is essential that the web server administrator create a basic index page that does not include any default product details BEFORE the server is connected to the internet. This will help prevent Google hackers from easily identifying our operating systems, and other running software.
Another popular search strategy that Google hackers use, will query for error messages from applications, such as the errors generated by a SQL database. As SQL injection is among the most popular attack techniques today, we can attempt to control the SQL information Google knows about by cleansing the input data from our applications.[20]

Removing content from Google indexes

Chances are that after Google hacking your site, you have found something you would like to remove from Google’s index. Google’s policy regarding the removal of links is “Google stops indexing the pages on your site only at the request of the webmaster who is responsible for those pages or as required by law. This policy is necessary to ensure that pages are not inappropriately removed from our index.”[21] While this is good from a freedom of speech standpoint, it doesn’t really help a security professional who needs to remove a breach of confidentiality.
Google does offer many different removal options depending on the circumstances. The first step in removal is to use the META tag techniques listed above. Google will not remove any links that do not block robots from scanning the site. Next, simply navigate to http://www.google.com/remove.html or http://www.google.com/intl/en/webmasters/3.html#removed, fill out the forms, and the next time Google crawls the internet, your content will not be indexed. This technique is not immediate, and it may take several weeks for Google to complete its crawl, and your content to be removed.
If you need to immediately remove something from the Google index, you must first register with Google’s removal system located at http://services.google.com:8882/urlconsole/controller?cmd=reload&lastcmd=login. Once you register, your removal options are to remove pages using either meta tags or the robots.txt file methods. You must provide the link to the robots.txt file, or file containing the meta tag information. The removal will take place immediately and will remain in effect as long as the meta tag or robots.txt file exists.
Google will immediately remove any content that is a violation of the Digital Millennium Copyright Act (DCMA). Follow the instructions at http://www.google.com/dmca.html if you need to remove protected content.

Google Hack Honeypot

20 http://www.spidynamics.com/support/whitepapers/WhitepaperSQLInjection.pdf

21 http://www.google.com/remove.html

The Google Hack Honeypot (GHH) is the reaction to malicious Google Hacking.[22] The GHH is written using the GHDB and provides a honeypot to many signatures contained in the GHDB. When a Google hacker performs a search the GHH project has a honeypot for, one of the results displayed is the honeypot. If the Google hacker clicks on the honeypot link, the honeypot writes the hacker's information to a log. If your organization subscribes to the honeypot theory, a GHH that mirrors production applications can provide valuable information on Google hackers that are targeting your site.

Going beyond Google

"One Google to rule them all with broad might, and americanocentric censorship, one Teoma to refine and refine and refine the queryes, one Fast for the obscure deep web, to bring us all in the depth, and in the darkness enlighten us..."[23] This paper has focused on the abilities of Google, but as http://searchlore.org/ explains, there are many other options that are just as powerful, in given situations. Searchlore.org details advanced options on many search engines, but concentrates on three main engines; Google.com, Teoma.com, and alltheweb.com or Fast. The Teoma search engine excels at suggesting refinements to queries, while Fast is the best at deep probing searches. Both Teoma and Fast offer different advanced options, and search strategies can be modified in the same manner as Google. Some of the automated tools outlined above can be modified to work with these other engines. Adding the capabilities of Teoma and Fast to your Google hacking practice will give you the most complete view of your company's internet presence

Conclusion

The popularity of Google hacking has grown by leaps and bounds over the last year. Many prominent security professionals have dedicated much of their time to speaking on the topic, as well as writing tools to further the practice. Worms have been written to use Google as a target locator, and popular open source communities have written honeypot software to attempt to gather information on Google hacking.

This paper has shown that many sites on the internet inadvertently give away more information than they intend to. It has suggested search strategies that companies can modify to Google hack themselves to look for this information, and tools to automate these searches.

Finally this paper has suggested methods to begin the defense against malicious Google hackers. The application development process has been examined to determine where Google hacking fits in.

Google hacking will exist as long as web applications and websites continue to

22 http://ghh.sourceforge.net/

23 Fravia, http://searchlore.org/main.htm#ring

deploy insecure solutions. Every security professional must take it upon themselves to Google hack their web presence on a regular basis to ensure their confidential information remains so.

List of References

The following materials were consulted in the preparation of this document.

Johnny Long. “Google Hacking Database”
http://johnny.ihackstuff.com/index.php?module=prodreviews

Raymond, Eric S. “Hacker” 29 Dec 2003
http://catb.org/~esr/jargon/html/H/hacker.html

“Google Corporate Information: Google History”
http://www.google.com/corporate/history.html

“Google Corporate Information: Technology”
http://www.google.com/corporate/tech.html

“Benefits of Google Search” http://www.google.com/technology/whyuse.html

Calishan, Tara, Rael Dornfest, and DJ Adams.. Google Pocket Guide; O’Reilly & Associates, 2003

Long, Johnny. Shmoocon 2005 presentation
http://johnny.ihackstuff.com/security/premium/2005_Shmoocon.pdf

“Google hacking database entry 467”
http://johnny.ihackstuff.com/index.php?module=prodreviews&func=showcontent&id=467

“Files containing passwords”
http://johnny.ihackstuff.com/index.php?module=prodreviews&func=reviewsbycat&reviewsel=13

Lemos, Rob. “Net worm using Google to spread” 22 Dec 2004
http://www.zdnet.com.au/news/security/0,2000061744,39175010,00.htm

Roberts, Paul. “New worm, Santy.A, using Google to spread” 21 Dec 2004
http://www.computerworld.com/securitytopics/security/story/0,10801,98453,00.html

SANS Institute. Track 1 – SANS Security Essentials. Volume 1.2. SANS Press, Jan 2004

SANS Institute. Track 1 – SANS Security Essentials. Volume 1.2. SANS Press, Jan 2004

"Special Publication 800-12: Chapter FIVE" 1 Jul 2004
http://csrc.nist.gov/publications/nistpubs/800-12/800-12-html/chapter5.html

SANS Institute. Track 1 – SANS Security Essentials. Volume 1.2. SANS Press, Jan 2004

Koster, Martijn. "A Standard for Robot Exclusion"
http://www.robotstxt.org/wc/norobots.html

Koster, Martijn. "The Web Robots FAQ"
http://www.robotstxt.org/wc/faq.html#log

Spett, Kevin. "SQL Injectionyou're your web applications vulnerable?" 2002
http://www.spidynamics.com/support/whitepapers/WhitepaperSQLInjection.pdf

"Remove Content from Google's Index" 2004
http://www.google.com/remove.html

"GHH – The "Google Hack" Honeypot" 17 Feb 2005 http://ghh.sourceforge.net/

Fravia. "main search engines' lore, usage and comparison" Jan 2005
http://searchlore.org/main.htm#ring