Chapter 6

Locating Exploits and Finding Targets

Solutions in this chapter:

- **Locating Exploit Code**
- **Locating Vulnerable Targets**
- **Links to Sites**

- ☑ **Summary**
- ☑ **Solutions Fast Track**
- ☑ **Frequently Asked Questions**

Introduction

Exploits, are tools of the hacker trade. Designed to penetrate a target, most hackers have many different exploits at their disposal. Some exploits, termed *zero day* or *0day*, remain underground for some period of time, eventually becoming public, posted to newsgroups or Web sites for the world to share. With so many Web sites dedicated to the distribution of exploit code, it's fairly simple to harness the power of Google to locate these tools. It can be a slightly more difficult exercise to locate potential targets, even though many modern Web application security advisories include a Google search designed to locate potential targets.

In this chapter we'll explore methods of locating exploit code and potentially vulnerable targets. These are not strictly "dark side" exercises, since security professionals often use public exploit code during a vulnerability assessment. However, only black hats use those tools against systems without prior consent.

Locating Exploit Code

Untold hundreds and thousands of Web sites are dedicated to providing exploits to the general public. Black hats generally provide exploits to aid fellow black hats in the hacking community. White hats provide exploits as a way of eliminating false positives from automated tools during an assessment. Simple searches such as *remote exploit* and *vulnerable exploit* locate exploit sites by focusing on common lingo used by the security community. Other searches, such as *inurl:0day,* don't work nearly as well as they used to, but old standbys like *inurl:sploits* still work fairly well. The problem is that most security folks don't just troll the Internet looking for exploit caches; most frequent a handful of sites for the more mainstream tools, venturing to a search engine only when their bookmarked sites fail them. When it comes time to troll the Web for a specific security tool, Google's a great place to turn first.

Locating Public Exploit Sites

One way to locate exploit code is to focus on the file extension of the source code and then search for specific content within that code. Since source code is the text-based representation of the difficult-to-read machine code, Google is well suited for this task. For example, a large number of exploits are written in C, which generally uses source code ending in a .c extension. Of course, a search for *filetype:c c* returns nearly 500,000 results, meaning that we need to narrow our search. A query for *filetype:c exploit* returns around 5,000 results, most of which are exactly the types of programs we're looking for. Bearing in mind that these are the most popular sites hosting C source code containing the word *exploit*, the returned list is a good start for a list of bookmarks. Using page-scraping techniques, we can isolate these sites by running a UNIX command such as:

```
grep Cached exploit_file  | awk -F" -" '{print $1}' | sort -u
```

against the dumped Google results page. Using good, old-fashioned cut and paste or a command such as *lynx –dump* works well for capturing the page this way. The slightly polished results of scraping 20 results from Google in this way are shown in the list below.

download2.rapid7.com/r7-0025
securityvulns.com/files
www.outpost9.com/exploits/unsorted
downloads.securityfocus.com/vulnerabilities/exploits
packetstorm.linuxsecurity.com/0101-exploits
packetstorm.linuxsecurity.com/0501-exploits
packetstormsecurity.nl/0304-exploits
www.packetstormsecurity.nl/0009-exploits
www.0xdeadbeef.info
archives.neohapsis.com/archives/
packetstormsecurity.org/0311-exploits
packetstormsecurity.org/0010-exploits
www.critical.lt
synnergy.net/downloads/exploits
www.digitalmunition.com
www.safemode.org/files/zillion/exploits
vdb.dragonsoft.com.tw
unsecure.altervista.org
www.darkircop.org/security
www.w00w00.org/files/exploits/

Underground Googling…

Google Forensics

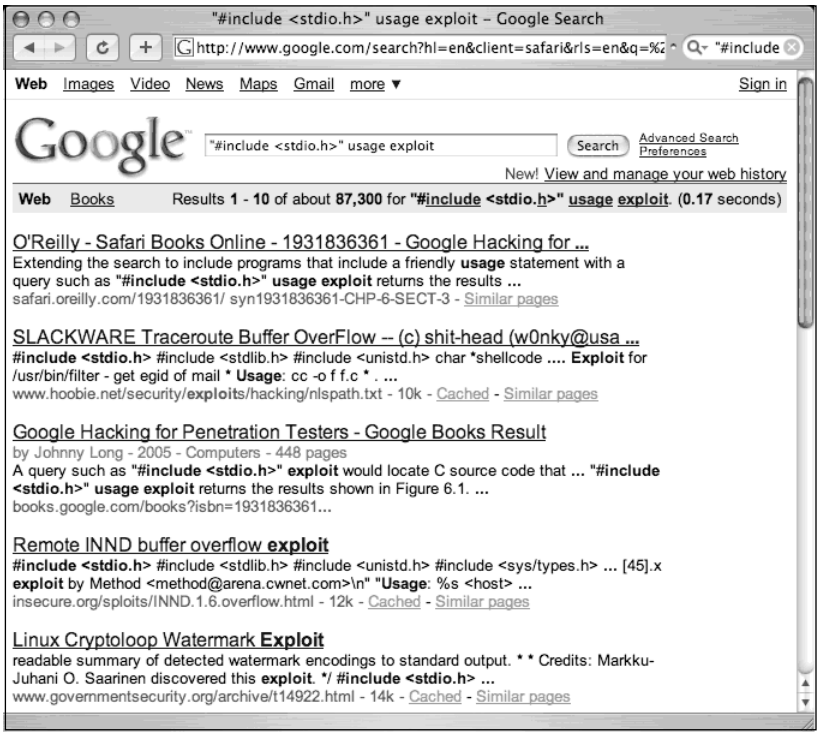
Google also makes a great tool for performing digital forensics. If a suspicious tool is discovered on a compromised machine, it's pretty much standard practice to run the tool through a UNIX command such as *strings –8* to get a feel for the readable text in the program. This usually reveals information such as the usage text for the tool, parts of which can be tweaked into Google queries to locate similar tools. Although obfuscation programs are becoming more and more commonplace, the combination of *strings* and Google is very powerful, when used properly—capable of taking some of the mystery out of the vast number of suspicious tools on a compromised machine.

Locating Exploits Via Common Code Strings

Since Web pages display source code in various ways, a source code listing could have practically any file extension. A PHP page might generate a text view of a C file, for example, making the file extension from Google's perspective .PHP instead of .C.

Another way to locate exploit code is to focus on common strings within the source code itself. One way to do this is to focus on common inclusions or header file references. For example, many C programs include the standard input/output library functions, which are referenced by an *include* statement such as *#include <stdio.h>* within the source code. A query such as *"#include <stdio.h>" exploit* would locate C source code that contained the word *exploit,* regardless of the file's extension. This would catch code (and code fragments) that are displayed in HTML documents. Extending the search to include programs that include a friendly usage statement with a query such as *"#include <stdio.h>" usage exploit* returns the results shown in Figure 6.1.

Figure 6.1 Searching for Exploit Code with Nonstandard Extensions



This search returns quite a few hits, nearly all of which contain exploit code. Using traversal techniques (or simply hitting up the main page of the site) can reveal other exploits or tools. Notice that most of these hits are HTML documents, which our previous *filetype:c*

query would have excluded. There are lots of ways to locate source code using common code strings, but not all source code can be fit into a nice, neat little box. Some code can be nailed down fairly neatly using this technique; other code might require a bit more query tweaking. Table 6.1 shows some suggestions for locating source code with common strings.

Table 6.1 Locating Source Code with Common Strings

Language	Extension (Optional)	Sample String
asp.net (C#)	Aspx	*"<%@ Page Language="C#"" inherits*
asp.net (VB)	Aspx	*"<%@ Page Language="vb"" inherits*
asp.net (VB)	Aspx	*<%@ Page LANGUAGE="JScript"*
C	C	*"#include <stdio.h>"*
C#	Cs	*"using System;" class*
c++	Cpp	*"#include "stdafx.h""*
Java	J, JAV	*class public static*
JavaScript	JS	*"<script language="JavaScript">"*
Perl	PERL, PL, PM	*"#!/usr/bin/perl"*
Python	Py	*"#!/usr/bin/env"*
VBScript	.vbs	*"<%@ language="vbscript" %>"*
Visual Basic	Vb	*"Private Sub"*

In using this table, a *filetype* search is optional. In most cases, you might find it's easier to focus on the sample strings so that you don't miss code with funky extensions.

Locating Code with Google Code Search

Google Code Search (www.google.com/codesearch) can be used to search for public source code. In addition to allowing queries that include powerful regular expressions, code search introduces unique operators, some of which are listed in Table 6.2.

Table 6.2 Google Code Search Operators

Operator	Description	Example
file	Search for specific types of files. Parameters can include file names, extensions, or full path names.	file:js
package	Search within a specific package, often listed as a URL or CVS server name	package:linux.*.tar.gz buggy

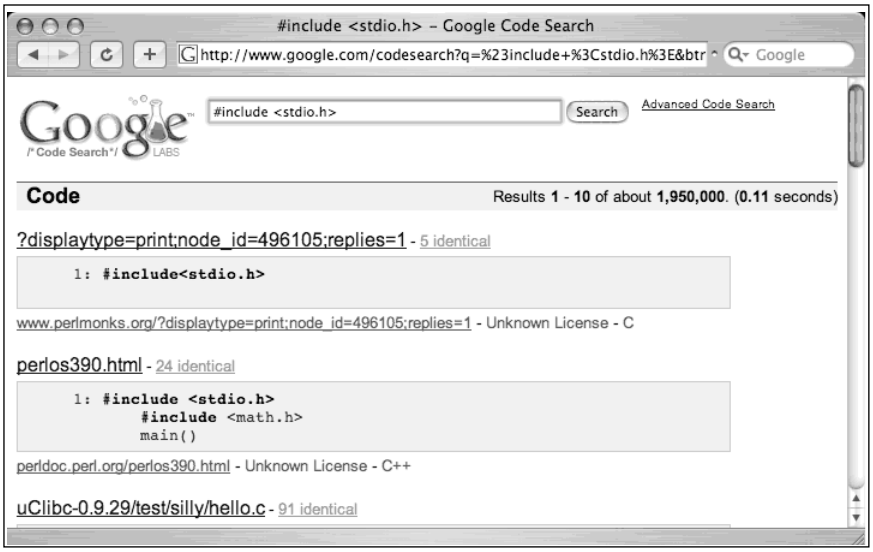
Continued

Table 6.2 Google Code Search Operators

Operator	Description	Example
lang	Search for code written in specific languages	lang:"c++"
license	Search for code written under specific licenses	license:gpl

Code search is a natural alternative to the techniques we covered in the previous section. For example, in Table 6.1 we used the web search term "*#include <stdio.h>*" to locate programs written in the *C* programming language. This search is effective, and locates *C* code, regardless of the file extension. This same query could be reformatted as a code search query by simply removing the quotes as shown in Figure 6.2.

Figure 6.2 Code Search used to locate Header Strings



If we're trying to locate *C* code, it makes more sense to query code search for *lang:c* or *lang:c++*. Although this may feel an awful lot like searching by file extension, this is a bit more advanced than a file extension search. Google's Code Search does a decent job of analyzing the code (regardless of extension) to determine the programming language the code was written in. Check out the second hit in Figure 6.2. As the snippet clearly shows, this is C code, but is embedded in an HTML file, as revealed by the file name, *perlos390.html*.

As many researchers and bloggers have reported, Google Code Search can also be used to locate software that contains potential vulnerabilities, as shown in Table Table 6.3.

Table 6.3 Google Code Searches for Vulnerable Code

Google Code Search Query	Description	Author
lang:php (echo|print).*\$_(GET|POST|COOKIE|REQUEST)	Code which displays untrusted variables passed GET/POST or cookies. Classic XSS (Cross-Site scripting) vulnerability.	Ilia Alshanetsky
<%=.*getParameter*	Code that allows XSS in Java due to HTML-encoded user input.	Nitesh Dhanjani
lang:php echo.*\$_SERVER\['PHP_SELF']	XSS vulnerability due to echo of PHP_SELF.	
echo.*\$_(GET|POST).*	Generic version of above query.	Chris Shiflett
lang:php query\(.*\$_(GET|POST|COOKIE|REQUEST).*\)	SQL queries built from user-supplied GET/POST requests. This could be an SQL injection point.	Ilia Alshanetsky
.*mysql_query\(.*\$_(GET|POST).*	SQL queries built from user-supplied GET/POST requests. This could be an SQL injection point. MySQL-specific.	Nitesh Dhanjani
lang:php "WHERE username='$_"	SQL injection due to raw input to WHERE clause.	Chris Shiflett
.*executeQuery.*getParameter.*	SQL injection in Java code due to execution of an SQL query executed with untrusted user input.	Stephen de Vries

Continued

Table 6.3 continued Google Code Searches for Vulnerable Code

Google Code Search Query	Description	Author
lang:php header\s*\("Location:.*\$_(GET|POST|COOKIE|REQUEST).*\)	Code import built from user-supplied GET/POST requests and cookies. This may allow execution of malicious code.	Ilia Alshanetsky
lang:php (system|popen|shell_exec|exec)\s*\(\$_(GET|POST|COOKIE|REQUEST).*\)	Code that passes untrusted GET/POST/COOKIE data to the system for execution. This allows remote code execution.	Ilia Alshanetsky

Locating Malware and Executables

Since the first edition of this book was published, researchers discovered that Google not only crawls, but analyzes *binary,* or *executable* files. The query *"Time Date Stamp: 4053c6c2"* (shown in Figure 6.3) returns one hit for a program named *Message.pif.* A PIF (or Program Information File) is a type of Windows executable.

Since executable files are machine (and not human) readable, it might seem odd to see text in the snippet of the search result. However, the snippet text is the result of Google's *analysis* of the binary file. Clicking the *View as HTML* link for this result displays the full analysis of the file, as shown in Figure 6.4. If the listed information seems like hardcore geek stuff, it's because the listed information is hardcore geek stuff.

Figure 6.3 Google Digs into Executable Files

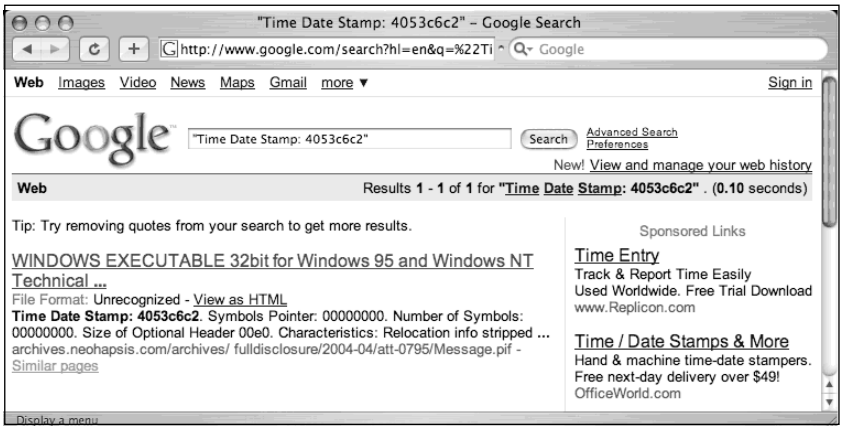
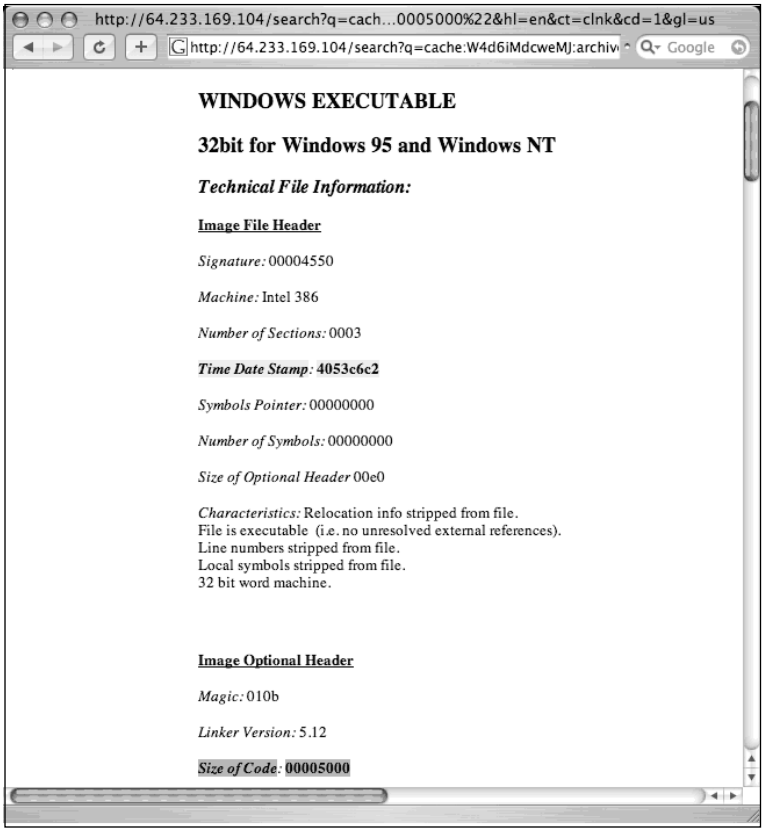


Figure 6.4 Google Analyzes Binary Files



Clicking the file link (instead of the HTML link) will most likely freak out your browser, as shown in Figure 6.5.

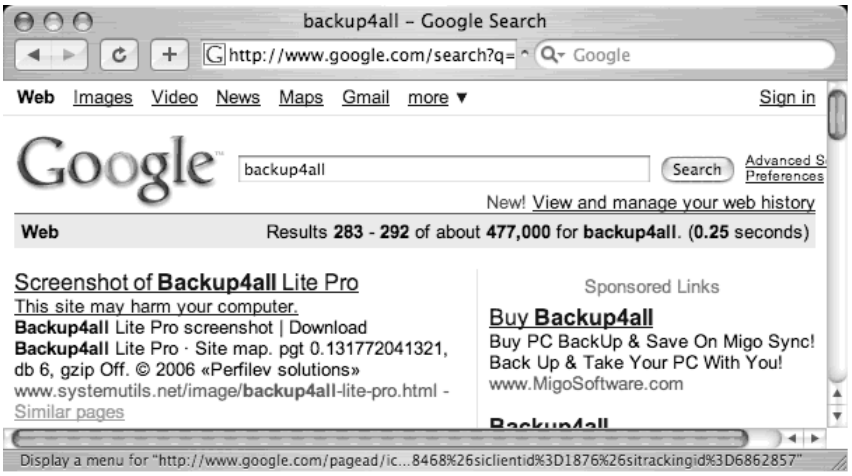
Figure 6.5 Binary Browser Garbage

Binary files were just not meant to be displayed in a browser. However, if we right-click the file link and choose *Save As…* to save it to our local machine, we can run our own basic analysis on the file to determine exactly what it is. For example, running the *file* command on a Linux or Mac OS X machine reveals that Message.pif is indeed a Windows Executable file:

```
$ file Message.pif.txt
Message.pif.txt: MS Windows PE 32-bit Intel 80386 GUI executable not relocatable
```

So Google snatches and analyzes binary files it finds on the web. So what? Well, first, it's interesting to see that Google has moved into this space. It's an indication that they're expanding their capabilities. For example, Google now has the ability to recognize malware. Consider the search for *Backup4all* backup software shown in Figure 6.6.

Figure 6.6 Google Warning about Malware

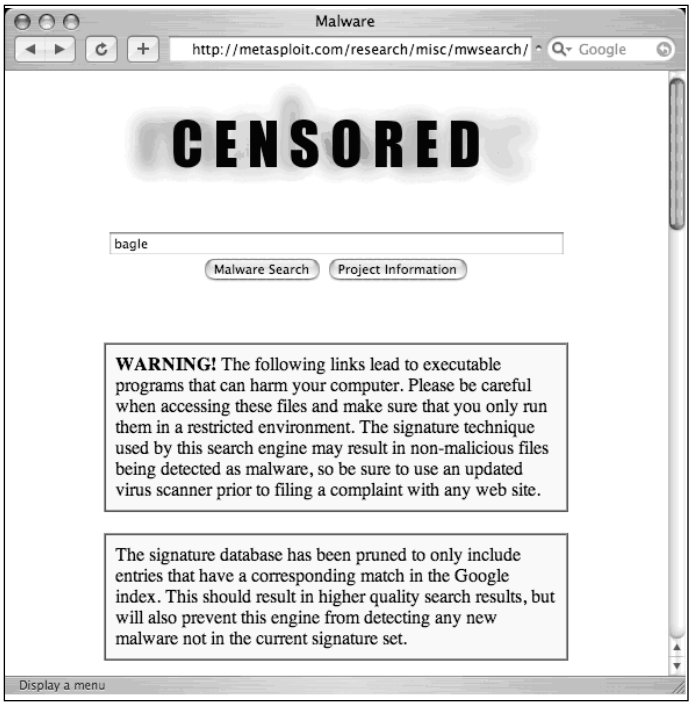


Notice the warning below the site description: This site may harm your computer. Clicking on the file link will not take you to the systemutils.net URL, but will instead present a warning page as show in Figure 6.7.

Figure 6.7 Google's Malware Wrapping Page

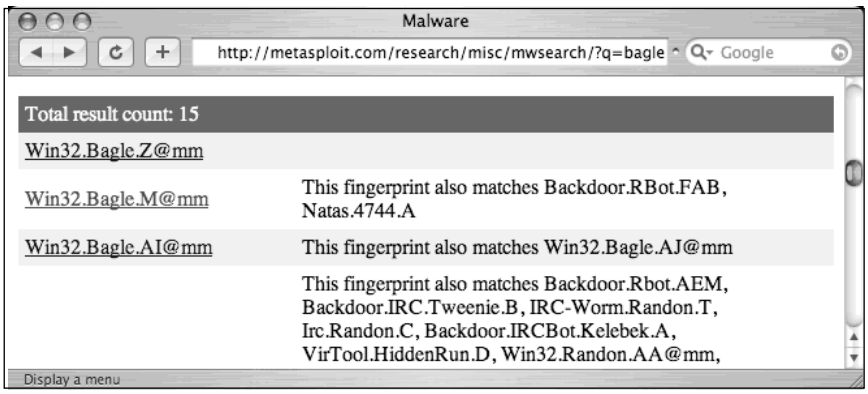
So this is certainly a handy feature, but since this book is about Google Hacking, not about Google's plans to save the world's Internet surfers from themselves, it's only right that we get to the dark heart of the matter: Google can be used to *search* for live malware. As Websense announced in 2006, this feature can be leveraged to search for very specific executables by focusing on specific details of individual files, such as the *Time Stamp, Size* and *Entry Point* fields. H.D. Moore took this one step further and created a sort of malware search engine, which can be found at http://metasploit.com/research/misc/mwsearch, as shown in Figure 6.8.

Figure 6.8 H.D. Moore's Malware Search Engine based on Google Binary Search



A search for *bagle*, for example, reveals several hits, as shown in Figure 6.9.

Figure 6.9 A Malware Search for Bagles (With No Cream Cheese)



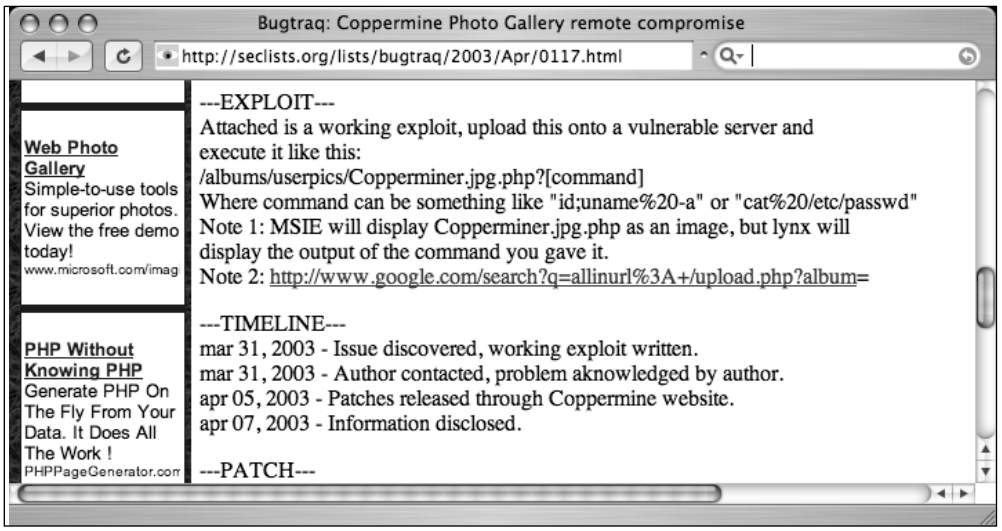
Clicking the second link in this search result will forward you to a Google web search results page for *"Time Date Stamp: 4053c6c2" "Size of Image: 00010000" "Entry Point: 0000e5b0" "Size of Code: 00005000"*—a very long query that uniquely describes the binary signature for the Win32.Bagle.M worm. The Google results page for this query is shown in Figure 6.3. Remember this file? It's the one we successfully downloaded and plopped right onto our desktop!

So even though Google's binary analysis capability has the potential for good, skillful attackers can use it for malicious purposes as well.

Locating Vulnerable Targets

Attackers are increasingly using Google to locate Web-based targets vulnerable to specific exploits. In fact, it's not uncommon for public vulnerability announcements to contain Google links to potentially vulnerable targets, as shown in Figure 6.10.

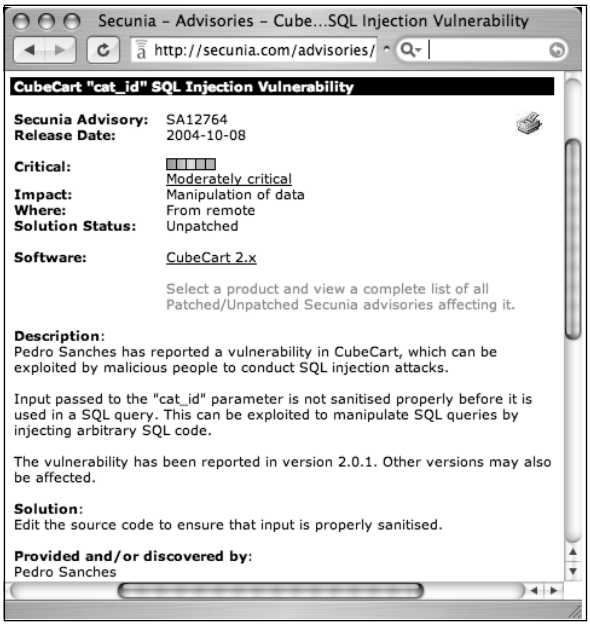
Figure 6.10 Google Link to Vulnerable Targets in Advisory



Locating Targets Via Demonstration Pages

The process of locating vulnerable targets can be fairly straightforward, as we'll see in this section. Other times, the process can be a bit more involved, as we'll see in the next section. Let's take a look at a Web application security advisory posted to Secunia (www.secunia.com) on October 10, 2004, as shown in Figure 6.11.

Figure 6.11 Typical Web Application Security Advisory



This particular advisory displays a link to the affected software vendor's Web site. Not all advisories list such a link, but a quick Google query should help you locate the vendor's page. Since our goal is to develop a query string to locate vulnerable targets on the Web, the vendor's Web site is a good place to discover what exactly the product's Web pages look like. Like many software vendors' Web sites, the CubeCart site shows links for product demonstrations and live sites that are running the product, as shown in Figure 6.12.

Figure 6.12 Vendor Web Pages Often Provide Product Demonstrations

At the time of this writing, this site's demonstration pages were offline, but the list of live sites was active. Live sites are often better for this purpose because we can account for potential variations in how a Web site is ultimately displayed. For example, some administrators might modify the format of a vendor-supplied Web page to fit the theme of the site. These types of modifications can impact the effectiveness of a Google search that targets a vendor-supplied page format.

Perusing the list of available live sites in Figure 6.4, we find that most sites look very similar and that nearly every site has a "powered by" message at the bottom of the main page, as shown in the (highly edited) example in Figure 6.13.

Figure 6.13 "Powered by" Tags Are Common Query Fodder for Finding Web Apps

In this case, the live page displays "Powered by CubeCart 2.0.1" as a footer on the main page. Since CubeCart 2.0.1 is the version listed as vulnerable in the security advisory, we need do little else to create a query that locates vulnerable targets on the Web. The final query, *"Powered by CubeCart 2.0.1",* returns results of over 27,000 potentially vulnerable targets, as shown in Figure 6.14.

Combining this list of sites with the exploit tool released in the Secunia security advisory, an attacker has access to a virtual smorgasbord of online retailers that could likely be compromised, potentially revealing sensitive customer information such as address, products purchased, and payment details.

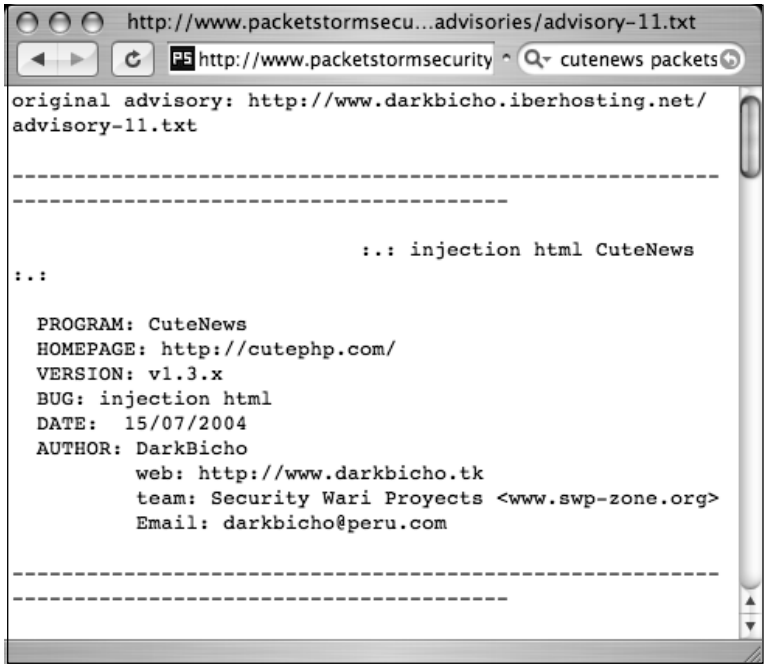
Figure 6.14 A Query That Locates Vulnerable CubeCart Sites

Locating Targets Via Source Code

In some cases, a good query is not as easy to come by, although as we'll see, the resultant query is nearly identical in construction. Although this method is more drawn out (and could be short-circuited by creative thinking), it shows a typical process for detecting an exact working query for locating vulnerable targets. Here we take a look at how a hacker might use the source code of a program to discover ways to search for that software with Google. For example, an advisory was released for the CuteNews program, as shown in Figure 6.15.

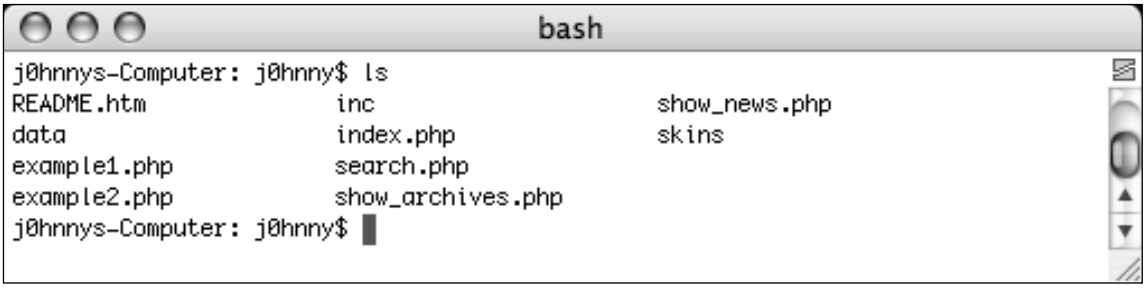
As explained in the security advisory, an attacker could use a specially crafted URL to gain information from a vulnerable target. To find the best search string to locate potentially vulnerable targets, we can visit the Web page of the software vendor to find the source code of the offending software. In cases where source code is not available, an attacker might opt to simply download the offending software and run it on a machine he controls to get ideas for potential searches. In this case, version 1.3.1 of the CuteNews software was readily available for download from the author's Web page.

Figure 6.15 The CuteNews Advisory



Once the software is downloaded and optionally unzipped, the first thing to look for is the main Web page that would be displayed to visitors. In the case of this particular software, PHP files are used to generate Web pages. Figure 6.16 shows the contents of the top-level CuteNews directory.

Figure 6.16 Files Included with CuteNews 1.3.1



Of all the files listed in the main directory of this package, index.php is the most likely candidate to be a top-level page. Parsing through the index.php file, line 156 would most likely catch our eye.

```
156 // If User is Not Logged In, Display The Login Page
```

Line 156 shows a typical informative comment. This comment reveals the portion of the code that would display a login page. Scrolling down farther in the login page code, we come to lines 173–178:

```
173         <td width=80>Username: </td>
174         <td><input tabindex=1 type=text
            name=username value='$lastusername' style=\"width:134\"></td>
175         </tr>
176         <tr>
177         <td>Password: </td>
178         <td><input type=password name=password style=\"width:134\"></td>
```

These lines show typical HTML code and reveal username and password prompts that are displayed to the user. Based on this code, a query such as *"username:" "password:"* would seem reasonable, except for the fact that this query returns millions of results that are not even close to the types of pages we are looking for. This is because the colons in the query are effectively ignored and the words *username* and *password* are far too common to use for even a base search. Our search continues to line 191 of index.php, shown here:

```
191 echofooter();
```

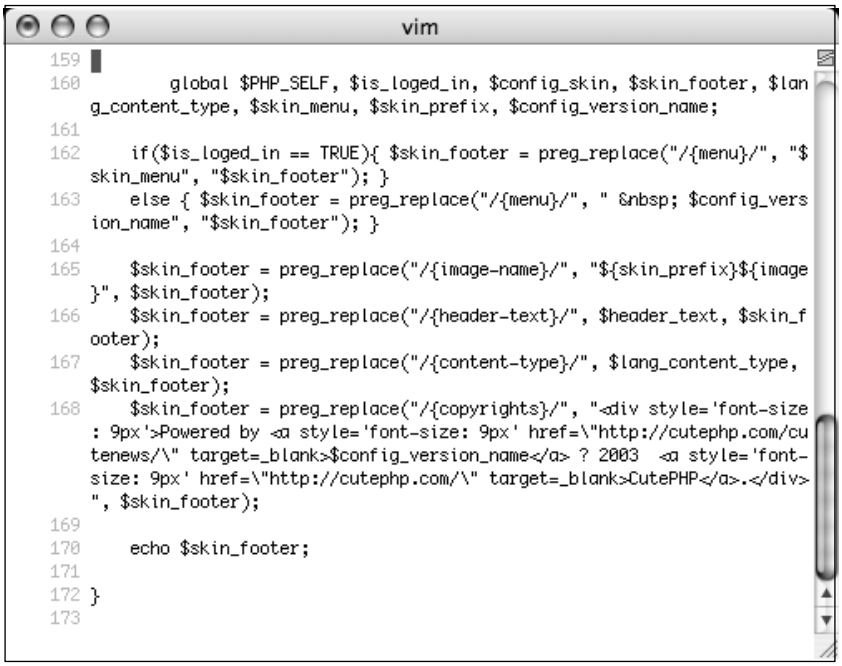
This line prints a footer at the bottom of the Web page. This line is a function, an indicator that it is used many times through the program. A common footer that displays on several CuteNews pages could make for a very nice base query. We'll need to uncover what exactly this footer looks like by locating the code for the *echofooter* function. Running a command such as *grep –r echofooter* * will search every file in each directory for the word *echofooter*. This returns too many results, as shown in this abbreviated output:

```
j0hnnys-Computer: j0hnny$ grep -r echofooter *
inc/about.mdu:  echofooter();
inc/addnews.mdu:    echofooter();
inc/categories.mdu:echofooter();
inc/editnews.mdu:    echofooter();
inc/editnews.mdu:    echofooter();
inc/editusers.mdu:    echofooter();
inc/functions.inc.php:  echofooter();
inc/functions.inc.php:// Function:      echofooter
inc/functions.inc.php:function echofooter(){
inc/help.mdu:   echofooter();
```

Most of the lines returned by this command are *calls* to the *echofooter* function, not the definition of the function itself. One line, however, precedes the word *echofooter* with the word *function*, indicating the definition of the function. Based on this output, we know that the file inc/functions.inc.php contains the code to print the Web page footer. Although

there is a great deal of information in this function, as shown in Figure 6.17, certain things will catch the eye of any decent Google hacker. For example, line 168 shows that copyrights are printed and that the term "Powered by" is printed in the footer.

Figure 6.17 The *echofooter* Function Reveals Potential Query Strings

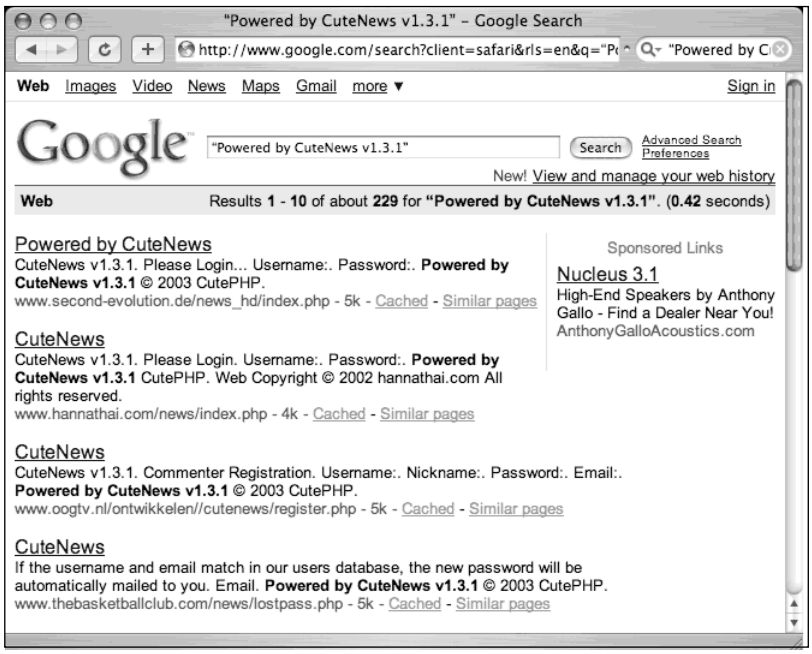


A phrase like "Powered by" can be very useful in locating specific targets due to their high degree of uniqueness. Following the "Powered by" phrase is a link to http://cutephp.com/cutenews/ and the string *$config_version_name*, which will list the version name of the CuteNews program. To have a very specific "Powered by" search to feed Google, the attacker must either guess the exact version number that would be displayed (remembering that version 1.3.1 of CuteNews was downloaded) or the actual version number displayed must be located in the source code. Again, *grep* can quickly locate this string for us. We can either search for the string directly or put an equal sign (=) after the string to find where it is defined in the code. A *grep* command such as *grep –r "\$config_version_name ="* * will do the trick:

```
johnny-longs-g4 root$ grep -r "\$config_version_name =" *
inc/install.mdu:\$config_version_name = "CuteNews v1.3.1";
inc/options.mdu:    fwrite($handler, "<?PHP \n\n//System
Configurations\n\n\$config_version_name =
\"$config_version_name\";\n\n\$config_version_id = $config_version_id;\n\n");
johnny-longs-g4 root$
```

As shown here, the version name is listed as *CuteNews v1.3.1*. Putting the two pieces of the footer together creates a very specific string: *"Powered by CuteNews v1.3.1"*. This in turn creates a very nice Google query, as shown in Figure 6.18. This very specific query returns nearly perfect results, displaying nearly 500 sites running the potentially vulnerable version 1.3.1 of the CuteNews software.

Figure 6.18 A Completed Vulnerability Search



Too many examples of this technique are in action to even begin to list them all, but in the tradition of the rest of this book, Table 6.4 lists examples of some queries designed to locate targets running potentially vulnerable Web applications. These examples were all pulled from the Google Hacking Database.

Table 6.4 Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
inurl:custva.asp	EarlyImpact Productcart contains multiple vulnerabilities in versions YaBB Gold - Sp 1.3.1 and others.
"Powered by mnoGoSearch— free web search engine software"	Certain versions of mnGoSearch contain a buffer overflow vulnerability
intitle:guestbook "advanced guestbook 2.2 powered"	Advanced Guestbook v2.2 has an SQL injection vulnerability

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
filetype:asp inurl: "shopdisplayproducts.asp"	Versions of VP-ASP (Virtual Programming—ASP) contains multiple cross-site scripting attacks vulnerabilities
"Powered by: vBulletin * 3.0.1" inurl:newreply.php	vBulletin 3.01 does not correctly sanitize the input, allowing malicious code injection.
"Powered by Invision Power Board(U) v1.3 Final"	Invision Power Board v.13 Final has an SQL injection vulnerability in its 'ssi.php' script.
"powered by sphider" -exploit -ihackstuff -www.cs.ioc.ee	Versions of the sphider search engine script allow arbitrary remote code inclusion.
inurl:gotoURL.asp?url=	Asp Nuke version 1.2, 1.3, and 1.4 does not sanitize the input vars, creating an SQL injection problem.
inurl:comersus_message.asp	Certain versions of Comersus Open Technologies Comersus Cart have Multiple Vulnerabilities, including XSS.
ext:pl inurl:cgi intitle:"FormMail *" -"*Referrer" -"* Denied" -sourceforge -error -cvs -input	Certain versions of FormMail contain configuration problems and invalid referrer checks.
inurl:"dispatch.php?atknodetype" | inurl:class.at	Certain versions of Achievo allow remote code execution.
"Powered by Gallery v1.4.4"	Gallery v1.44 contains a vulnerability that may allow a remote attacker to execute malicious scripts
"Powered by Ikonboard 3.1.1"	IkonBoard 3.1.1 contains poor user input validation, allowing an attacker to evaluate arbitrary Perl and run arbitrary commands.
inurl:/cgi-bin/index.cgi inurl:topics inurl:viewca	Certain versions of WebAPP contain a serious reverse directory traversal vulnerability.
inurl:"/becommunity/community/index.php?pageurl="	Certain versions of E-market allow arbitrary code injection.
"Powered *: newtelligence" ("dasBlog 1.6"| "dasBlog 1.5"| "dasBlog 1.4"|"dasBlog 1.3")	DasBlog 1.3-1.6 is reportedly susceptible to an HTML injection.
"Powered by DCP-Portal v5.5"	DCP-Portal 5.5 is vulnerable to sql injection.
"FC Bigfeet" -inurl:mail	Certain versions of TYPO3 allow demo logins.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
filetype:cgi inurl:tseekdir.cgi	Certain versions of Turbo Seek allow for file enumeration.
filetype:php inurl:index.php inurl: "module=subjects" inurl:"func=*" (listpages| viewpage | listcat)	Certain versions of the PostNuke Modules Factory Subjects module contain an SQL injection vulnerability.
filetype:cgi inurl:pdesk.cgi	Certain versions of PerlDesk contain multiple vulnerabilities.
"Powered by IceWarp Software" inurl:mail	IceWarp Web Mail prior to v 5.2.8 contains multiple input validation vulnerabilities.
intitle:"MRTG/RRD" 1.1* (inurl:mrtg.cgi | inurl:14all.cgi |traffic.cgi)	MRTG v1.1.* allow partial file enumeration.
inurl:com_remository	Certain versions of the ReMOSitory module for Mambo are prone to an SQL injection vulnerability.
intitle:"WordPress > * > Login form" inurl:"wp-login.php"	Certain versions of WordPress contain XSS vulnerabilities.
inurl:"comment.php?serendipity"	Certain versions of Serendipity are vulnerable to SQL injection.
"Powered by AJ-Fork v.167"	AJ-Fork v.167 is vulnerable to a full path disclosure.
"Powered by Megabook *" inurl :guestbook.cgi	Certain versions of MegaBook are prone to multiple HTML injection vulnerabilities.
"Powered by yappa-ng"	Certain versions of yappa-ng contain an authentication vulnerability.
"Active Webcam Page" inurl:8080	Certain versions of Active WebCam contain directory traversal and XSS vulnerabilities.
"Powered by A-CART"	Certain versions of A-CART allow for the downloading of customer databases.
"Online Store - Powered by ProductCart"	Certain versions of ProductCart contain multiple SQL injection vulnerabilities.
"Powered by FUDforum"	Certain versions of FUDforum contain SQL injection problems and file manipulation problems.
"BosDates Calendar System " "powered by BosDates v3.2 by BosDev"	BosDates 3.2 has an SQL injection vulnerability.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
intitle:"EMUMAIL - Login" "Powered by EMU Webmail"	EMU Webmail version 5.0 and 5.1.0 contain XSS vulnerabilities.
intitle:"WebJeff - FileManager" intext:"login" intext:Pass|PAsse	WebJeff-Filemanager 1.x has a directory traversal vulnerability.
inurl:"messageboard/Forum.asp?"	Certain versions of GoSmart Message Board suffer from SQL injection and XSS problems.
"1999-2004 FuseTalk Inc" -site:fusetalk.com	Fusetalk forums v4 are susceptible to XSS attacks.
"2003 DUware All Rights Reserved"	Certain versions of multiple DUware products suffer from SQL injection and HTML injection.
"This page has been automatically generated by Plesk Server Administrator"	Certain versions of Plesk Server Administrator (PSA) contain input validation errors.
inurl:ttt-webmaster.php	Turbo traffic trader Nitro v1.0 suffers from multiple vulnerabilities.
"Copyright Ã‚Â© 2002 Agustin Dondo Scripts"	Certain versions of CoolPHP suffer from multiple vulnerabilities.
"Powered by CubeCart"	CubeCart 2.0.1 has a full path disclosure and SQL injection problem.
"Ideal BB Version: 0.1" -idealbb.com	Ideal BB 0.1 is reported prone to multiple unspecified input validation vulnerabilities.
"Powered by YaPig V0.92b"	YaPiG v0.92b is reported to contain an HTML injection vulnerability.
inurl:"/site/articles.asp?idcategory="	Certain versions of Dwc_Articles suffer from possible sql injections.
filetype:cgi inurl:nbmember.cgi	Certain versions of Netbilling nbmember.cgicontains an information disclosure vulnerability.
"Powered by Coppermine Photo Gallery"	Coppermine Photo Gallery Coppermine Photo Gallery 1.0, 1.1, 1.2, 1.2.1, 1.3, 1.3.1 and 1.3.2 contains a design error that may allow users to cast multiple votes for a picture.
"Powered by WowBB" -site:wowbb.com	Certain versions of WowBB are reportedly affected by multiple input validation vulnerabilities.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
"Powered by ocPortal" -demo -ocportal.com	Certain versions of ocPortal is affected by a remote file include vulnerability.
inurl:"slxweb.dll"	Certain versions of SalesLogix contain authentication vulnerability.
"Powered by DMXReady Site Chassis Manager" -site:dmxready.com	Certain versions of the DMXReady Site Chassis Manager are susceptible to two remotely exploitable input validation vulnerabilities.
"Powered by My Blog" intext: "FuzzyMonkey.org"	FuzzyMonkey My Blog versions 1.15-1.20 are vulnerable to multiple input validation vulnerabilities.
inurl:wiki/MediaWiki	MediaWiki versions 1.3.1-6 are reported prone to a cross-site scripting vulnerability. This issue arises due to insufficient sanitization of user-supplied data.
"inurl:/site/articles.asp?idcategory="	Dwc_Articles version prior to v1.6 suffers from SQL injection vulnerabilities.
"Enter ip" inurl:"php-ping.php"	Certain versions of php-ping may be prone to a remote command execution vulnerabilities.
intitle:welcome.to.horde	Certain versions of Horde Mail suffer from several vulnerabilities.
"BlackBoard 1.5.1-f | Ã‚Â© 2003-4 by Yves Goergen"	BlackBoard Internet Newsboard System v1.5.1is reported prone to a remote file include vulnerability.
inurl:"forumdisplay.php" +"Powered by: vBulletin Version 3.0.0..4"	vBulletin 3.0.0.4 is reported vulnerable to a remote SQL injection vulnerability.
inurl:technote inurl:main.cgi *filename=*	Certain versions of Technote suffer from a remote command execution vulnerability.
"running: Nucleus v3.1" -.nucleuscms.org -demo	Multiple unspecified vulnerabilities reportedly affect Nucleus CMS v3.1.
"driven by: ASP Message Board"	Infuseum ASP Message Board 2.2.1c suffers from multiple unspecified vulnerabilities.
"Obtenez votre forum Aztek" -site:forum-aztek.com	Certain versions of Atztek Forum are prone to multiple input validation vulnerabilities.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
intext:("UBB.threadsÃ¢â?žÂ¢ 6.2" |"UBB.threadsÃ¢â?žÂ¢ 6.3") intext: "You * not logged *" -site:ubbcentral.com	UBB.Threads 6.2.*-6.3.* contains a one character brute force vulnerability.
inurl:/SiteChassisManager/	Certain versions of DMXReady Site Chassis Manager suffer from SQL and XSS vulnerabilities.
inurl:directorypro.cgi	Certain versions of DirectoryPro suffer from directory traversal vulnerabilities.
inurl:cal_make.pl	Certain versions of PerlCal allows remote attackers to access files that reside outside the normally bounding HTML root directory.
"Powered by PowerPortal v1.3"	PowerPortal 1.3 is reported vulnerable to remote SQL injection.
"powered by minibb" -site:www.minibb.net -intext:1.7f	miniBB versions prior to 1.7f are reported vulnerable to remote SQL injection.
inurl:"/cgi-bin/loadpage.cgi?user_id="	Certain versions of EZshopper allow Directory traversal.
intitle:"View Img" inurl:viewimg.php	Certain versions of the 'viewing.php' script does not properly validate user-supplied input in the 'path' variable.
+"Powered by Invision Power Board v2.0.0.2"	Inivision Power Board v2.0.0-2.0.2 suffers from an SQL injection vulnerability.
+"Powered by phpBB 2.0.6..10" -phpbb.com -phpbb.pl	phpbb 2.0.6-20.10 is vulnerable to SQL Injection.
ext:php intext:"Powered by phpNewMan Version"	Certain versions of PHP News Manager are vulnerable to a directory traversal problem.
"Powered by WordPress" -html filetype:php -demo -wordpress.org -bugtraq	Certain versions of WordPress are vulnerable to a few SQL injection queries.
intext:Generated.by.phpix.1.0? inurl:$mode=album	PHPix v1.0 suffers from a directory traversal vulnerability.
inurl:citrix/metaframexp/default/ login.asp? ClientDetection=On	Certain versions of Citrix contain an XSS vulnerability in a widely used version of their Web Interface.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
"SquirrelMail version 1.4.4" inurl:src ext:php	SquirrelMail v1.4.4 contains an inclusion vulnerability.
"IceWarp Web Mail 5.3.0" "Powered by IceWarp"	IceWarp Web Mail 5.3.0 contains multiple cross-site scripting and HTML injection vulnerabilities.
"Powered by MercuryBoard [v1"	MercuryBoard v1 contains an unspecified vulnerability.
"delete entries" inurl: admin/delete.asp	Certain versions of AspJar contain a flaw that may allow a malicious user to delete arbitrary messages.
allintitle:aspjar.com guestbook	Certain versions of the ASPJar guestbook contain an input validation vulnerability.
"powered by CubeCart 2.0"	Brooky CubeCart v2.0 is prone to multiple vulnerabilities due to insufficient sanitization of user-supplied data.
Powered.by:.vBulletin.Version ...3.0.6	vBulletin 3.0.6 is reported prone to an arbitrary PHP script code execution vulnerability.
filetype:php intitle:"paNews v2.0b4"	PaNews v2.0b4 is reported prone to a remote PHP script code execution vulnerability.
"Powered by Coppermine Photo Gallery" ("v1.2.2 b" | "v1.2.1" | "v1.2" | "v1.1" | "v1.0")	Coppermine Photo Gallery versions 1.0, 1.1, 1.2, 1.2.1 and 1.2.2b are prone to multiple input validation vulnerabilities, some of which may lead to arbitrary command execution.
powered.by.instaBoard.version.1.3	InstaBoard v1.3 is vulnerable to SQL Injection.
intext:"Powered by phpBB 2.0.13" inurl:"cal_view_month.php"|inurl: "downloads.php"	phpBB 2.0.13 with installed Calendar Pro MOD are vulnerable to SQL injection attacks.
intitle:"myBloggie 2.1.1..2— by myWebland"	myBloggie v2.1.1-2.1.2 is affected by multiple vulnerabilities.
intitle:"osTicket :: Support Ticket System"	Certain versions of osTicket contains several vulnerabilities.
inurl:sphpblog intext:"Powered by Simple PHP Blog 0.4.0"	Simple PHP Blog v0.4.0 is vulnerable to multiple attacks including full path disclosure, XSS and other disclosures.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
intitle:"PowerDownload" ("PowerDownload v3.0.2 Ã‚Â©" | "PowerDownload v3.0.3 Ã‚Â©") -site:powerscripts.org	PowerDownload version 3.0.2 and 3.0.3 contains a remote execution vulnerability.
"portailphp v1.3" inurl:"index.php ?affiche" inurl:"PortailPHP" -site:safari-msi.com	PortailPHP v1.3 suffers from an SQL injection vulnerability.
+intext:"powered by MyBulletinBoard"	MyBB <= 1.00 RC4 contains an SQL injection vulnerability.
intext:"Powered by flatnuke-2.5.3" +"Get RSS News" -demo	FlatNuke 2.5.3 contains multiple vulnerabilities.
intext:"Powered By: Snitz Forums 2000 Version 3.4.00..03"	Snitz Forum 2000 v 3.4.03 and older are vulnerable to many things including XSS.
inurl:"/login.asp?folder=" "Powered by: i-Gallery 3.3"	i-Gallery 3.3 (and possibly older) are vulnerable to many things, including directory traversals.
intext:"Calendar Program Ã‚Â© Copyright 1999 Matt Kruse" "Add an event"	Certain versions of CalendarScript is vulnerable to HTML injection.
"powered by PhpBB 2.0.15" -site:phpbb.com	phpBB 2.0.15 Viewtopic.PHP contains a remote code execution vulnerability.
inurl:index.php fees shop link.codes merchantAccount	EPay Pro version 2.0 is vulnerable to a directory traversal issue.
intitle:"blog torrent upload"	Certain versions of Blog Torrent contain a password revelation issue.
"Powered by Zorum 3.5"	Zorum 3.5 contains a remote code execution vulnerability.
"Powered by FUDForum 2.6" -site:fudforum.org -johnny.ihackstuff	FUDforum 2.6 is prone to a remote arbitrary PHP file upload vulnerability.
intitle:"Looking Glass v20040427" "When verifying	Looking Glass v20040427 allows arbitrary commands execution and cross site scripting.
phpLDAPadmin intitle: phpLDAPadmin filetype:php inurl: tree.php | inurl:login.php | inurl: donate.php (0.9.6 | 0.9.7)	phpLDAPadmin 0.9.6 - 0.9.7/alpha5 (and possibly prior versions) contains system disclosure, remote code execution, and XSS vulnerabilities.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
"powered by ITWorking"	SaveWebPortal 3.4 contains a remote code execution, admin check bypass and remote file inclusion vulnerability.
intitle:guestbook inurl:guestbook "powered by Adva	Certain versions of Advanced Guestbook are prone to HTML injection vulnerabilities.
"Powered by FUDForum 2.7" -site:fudforum.org -johnny.ihackstuff	FUDforum 2.7 is prone to a remote arbitrary PHP file upload vulnerability.
inurl:chitchat.php "choose graphic"	Cyber-Cats ChitCHat 2.0 contains multiple vulnerabilities.
"Calendar programming by AppIdeas.com" filetype:php	phpCommunityCalendar 4.0.3 (and possibly prior versions) allows SQL injection, login bypass and XSS.
"Powered by MD-Pro" | "made with MD-Pro"	MAXdev MD-Pro 1.0.73 (and possibly prior versions) allow remote code execution, XSS and path disclosure.
"Software PBLang" 4.65 filetype:php	PBLang 4.65 (and possibly prior versions) allow remote code execution, administrative credentials disclosure, system information disclosure, XSS and path disclosure.
"Powered by and copyright class-1" 0.24.4	Class-1 Forum Software v 0.24.4 allows remote code execution.
"Powered by AzDg" (2.1.3 | 2.1.2 | 2.1.1)	AzDGDatingLite V 2.1.3 (and possibly prior versions) allows remote code execution.
"Powered by: Land Down Under 800" | "Powered by: Land Down Under 801" - www.neocrome.net	Land Down Under 800 and 900 are prone to an HTML injection vulnerability.
"powered by Gallery v" "[slideshow]" |"images" inurl:gallery	Certain versions of Gallery suffer from a script injection vulnerability.
intitle:guestbook inurl:guestbook "powered by Advanced guestbook 2.*" "Sign the Guestbook"	Advanced Guestbook v2.* is prone to an HTML injection vulnerability.
"Copyright 2004 Ã‚Â© Digital Scribe v.1.4"	Digital Scribe v1.4 alows login bypass, SQL injection and remote code execution.
"Powered by PHP Advanced Transfer Manager v1.30"	PHP Advanced Transfer Manager v1.30 allows underlying system disclosure, remote command execution and cross site scripting.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
"Powered by CuteNews"	CuteNews 1.4.0 (and possibly prior versions) allows remote code execution.
"Powered by GTChat 0.95"+ "User Login"+"Remember my login information"	GTChat v0.95 contains a remote denial of service vulnerability.
intitle:"WEB//NEWS Personal Newsmanagement" intext:"Ã‚Â © 2002-2004 by Christian Scheb—Stylemotion.de"+"Version 1.4 "+ "Login"	WEB//NEWS 1.4 is prone to multiple SQL injection vulnerabilities.
"Mimicboard2 086"+"2000 Nobutaka Makino"+"password"+ "message" inurl:page=1	Mimicboard2 v086 is prone to multiple HTML injection vulnerabilities.
"Maintained with Subscribe Me 2.044.09p"+"Professional" inurl:"s.pl"	Subscribe Me Pro 2.0.44.09p is prone to a directory traversal vulnerability.
"Powered by autolinks pro 2.1" inurl:register.php	AutoLinksPro v2.1 contains a remote PHP File include vulnerability.
"CosmoShop by Zaunz Publishing" inurl:"cgi-bin/cosmoshop/lshop.cgi" -johnny.ihackstuff.com -V8.10.106 -V8.10.100 -V.8.10.85 -V8.10.108 -V8.11*	Cosmoshop versions 8.10.85, 8.10.100, 8.10.106, 8.10.108 and 8.11* are vulnerable to SQL injection, and cleartext password enumeration.
"Powered by Woltlab Burning Board" -"2.3.3" -"v2.3.3" -"v2.3.2" -"2.3.2"	Woltlab Burning Board versions 2.3.32 and 2.3.3 are vulnerable to SQL injection.
intitle:"PHP TopSites FREE Remote Admin"	Certain versions of PHP TopSites discloses configuration data to remote users.
Powered by PHP-Fusion v6.00.109 Ã‚Â© 2003-2005. -php-fusion.co.uk	PHP-Fusion v6.00.109 is prone to SQL Injection and administrative credentials disclosure.
"Powered By: lucidCMS 1.0.11"	Lucid CMS 1.0.11 has SQL injection and login bypass vulnerabilities.
"News generated by Utopia News Pro" | "Powered By: Utopia News Pro"	Utopia News Pro 1.1.3 (and prior versions) contain SQL Injection and XSS vulnerabilities.
intitle:Mantis "Welcome to the bugtracker" "0.15 | 0.16 | 0.17 | 0.18"	Mantis versions 0.19.2 or less contain XSS and SQL injection vulnerabilities.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
"Cyphor (Release:" -www.cynox.ch	Cyphor 0.19 (and possibly prior versions) allow SQL injection, board takeover and XSS.
"Welcome to the versatileBulletinBoard" | "Powered by versatileBulletinBoard"	VersatileBulletinBoard V1.0.0 RC2 (and possibly prior versions) contains multiple vulnerabilities.
inurl:course/category.php | inurl:course/info.php | inurl: iplookup/ipatlas/plot.php	Moodle <=1.6 allows blind SQL injection.
"Powered by XOOPS 2.2.3 Final"	XOOPS 2.2.3 allows arbitrary local file inclusion.
inurl:"wfdownloads/viewcat.php ?list="	XOOPS WF_Downloads (2.05) module allows SQL injection.
"This website was created with phpWebThings 1.4"	phpWebThings 1.4 contains several vulnerabilities.
"Copyright 2000 - 2005 Miro International Pty Ltd. All rights reserved" "Mambo is Free Software released"	Mambo 4.5.2x allows remote command execution.
("Skin Design by Amie of Intense")| ("Fanfiction Categories" "Featured Stories")|("default2, 3column, Romance, eFiction")	eFiction <=2.0 contains multiple vulnerabilities.
"Powered by UPB" (b 1.0)|(1.0 final)| (Public Beta 1.0b)	UPB versions b1.0, 1.0 final and Public Beta 1.0b Contains several vulnerabilities.
"powered by GuppY v4"|"Site crÃƒÂ©ÃƒÂ© avec GuppY v4"	Guppy <= 4.5.9 allows remote code execution and arbitrary inclusion.
"Powered by Xaraya" "Copyright 2005"	Xaraya <=1.0.0 RC4 contains a denial of service.
"This website powered by PHPX" -demo	PhpX <= 3.5.9 allows SQL injection and login bypass.
"Based on DoceboLMS 2.0"	DoceboLMS 2.0 contains multiple vulnerabilities.
"2005 SugarCRM Inc. All Rights Reserved" "Powered By SugarCRM"	Sugar Suite 3.5.2a & 4.0beta allow remote code execution.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
"Powered By phpCOIN 1.2.2"	PhpCOIN 1.2.2 allows arbitrary remote\local inclusion, blind SQL injection and path disclosure.
intext:"Powered by SimpleBBS v1.1"*	SimpleBBS v1.1 contains a flaw that may allow an attacker to carry out an SQL injection attack.
"Site powered By Limbo CMS"	Limbo Cms <= 1.0.4.2 allows remote code execution.
intext:"Powered by CubeCart 3.0.6" intitle:"Powered by CubeCart"	CubeCart 3.0.6 allows remote command execution.
intext:"PhpGedView Version" intext:"final - index" -inurl:demo	PHPGedView <=3.3.7 allows remote code execution.
intext:"Powered by DEV web management system" -dev-wms. sourceforge.net -demo	DEV cms <=1.5 allows SQL injection.
intitle:"phpDocumentor web interface"	Php Documentor < = 1.3.0 rc4 allows remote code execution.
inurl:install.pl intitle:GTchat	Certain versions of Gtchat allow unauthorized configuration changes.
intitle:"4images - Image Gallery Management System" and intext: "Powered by 4images 1.7.1"	4Images v1.7.1 allows remote code execution.
(intitle:"metaframe XP Login")| (intitle:"metaframe Presentation server Login")	Certain versions of Metaframe Presentation Server may allow unauthorized admin access.
"Powered by Simplog"	Simplog v1.0.2 allows directory traversal and XSS.
"powered by sblog" +"version 0.7"	Sblog v0.7 allows HTML injection.
"Thank You for using WPCeasy"	Certain versions of WPC.easy, allow SQL injection.
"Powered by Loudblog"	LoudBlog <= 0.4 contains an arbitrary remote inclusion vulnerability.
"This website engine code is copyright" "2005 by Clever Copy" -inurl:demo	Clever Copy <= 3.0 allows SQL injection.
"index of" intext:fckeditor inurl: fckeditor	FCKEditor script 2.0 and 2.2 contain multiple vulnerabilities.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
"powered by runcms" -runcms.com -runcms.org	Runcms versions <=1.2 are vulnerable to an arbitrary remote inclusion.
(intitle:"Flyspray setup"|"powered by flyspray 0.9.7") -flyspray.rocks.cc	Flyspray v0.9.7contains multiple vulnerabilities.
intext:"LinPHA Version" intext: "Have fun"	Linpha <=1.0 allows arbitrary local inclusion.
("powered by nocc" intitle:"NOCC Webmail") -site:sourceforge.net -Zoekinalles.nl -analysis	Certain versions of NOCC Webmail allow arbitrary local inclusion, XSS and possible remote code execution.
intitle:"igenus webmail login"	Igenus webmail allows local file enumeration.
"powered by 4images"	4images <= 1.7.1 allows remote code execution.
intext:"Powered By Geeklog" -geeklog.net	Certain versions of Geeklog contains multiple vulnerabilities.
intitle:admbook intitle:version filetype:php	Admbook version: 1.2.2 allows remote execution.
WEBalbum 2004-2006 duda -ihackstuff -exploit	WEBalbum 2004-2006 contains multiple vulnerabilities.
intext:"powered by gcards" -ihackstuff -exploit	Gcards <=1.45 contains multiple vulnerabilities.
"powered by php icalendar" -ihackstuff -exploit	php iCalendar <= 2.21 allows remote command execution.
"Powered by XHP CMS" -ihackstuff -exploit -xhp.targetit.ro	XHP CMS 0.5 allows remote command execution.
inurl:*.exe ext:exe inurl:/*cgi*/	Many CGI-bin executables allow XSS and html injection.
"powered by claroline" -demo	Claroline e-learning platform <= 1.7.4 contains multiple vulnerabilities.
"PhpCollab . Log In" | "NetOffice . Log In" | (intitle:"index.of." intitle: phpcollab|netoffice inurl:phpcollab |netoffice -gentoo)	PhpCollab 2.x / NetOffice 2.x allows SQL injection.
intext:"2000-2001 The phpHeaven Team" -sourceforge	PHPMyChat <= 0.14.5 contains an SQL injection vulnerability.
"2004-2005 ReloadCMS Team."	ReloadCMS <= 1.2.5stable allows XSS and remote command execution.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
intext:"2000-2001 The phpHeaven Team" -sourceforge	Certain versions of phpHeaven allow remote command execution.
inurl:server.php ext:php intext:"No SQL" -Released	Certain versions of PHPOpenChat contain multiple vulnerabilities.
intitle:PHPOpenChat inurl: "index.php?language="	Certain versions of PHPOpenchat allow SQL injection and information disclosure.
"powered by phplist" | inurl:" lists/?p=subscribe" | inurl:"lists/index. php?p=subscribe" -ubbi -bugs +phplist -tincan.co.uk	PHPList 2.10.2 allows arbitrary local file inclusion.
inurl:"extras/update.php" intext: mysql.php -display	Certain versions of osCommerce allow local file enumeration.
inurl:sysinfo.cgi ext:cgi	Sysinfo 1.2.1allows remote command execution.
inurl:perldiver.cgi ext:cgi	Certain versions of perldiver.cgi allow XSS.
inurl:tmssql.php ext:php mssql pear adodb -cvs -akbk	Certain versions of tmssql.php allow remote code execution.
"powered by php photo album" | inurl:"main.php?cmd=album" -demo2 -pitanje	Certain versions of PHP photo album allow local file enumeration and remote exploitation.
inurl:resetcore.php ext:php	Certain versions of e107 contain multiple vulnerabilities.
"This script was created by Php-ZeroNet" "Script. Php-ZeroNet"	Php-ZeroNet v 1.2.1 contains multiple vulnerabilities.
"You have not provided a survey identification num	PHP Surveyor 0995 allows SQL injection.
intitle:"HelpDesk" "If you need additional help, please email helpdesk at"	PHP Helpdesk 0.6.16 allows remote execution of arbitrary data.
inurl:database.php | inurl:info_ db.php ext:php "Database V2.*" "Burning Board *"	Woltlab Burning Board 2.x contains multiple vulnerabilities.
intext:"This site is using phpGraphy" | intitle:"my phpgraphy site"	phpGraphy 0911 allows XSS and denial of service.
intext:"Powered by PCPIN.com" -site:pcpin.com -ihackstuff -"works with" -findlaw	Certain versions of PCPIN Chat allow SQL injection, login bypass and arbitrary local inclusion.

Continued

Table 6.4 continued Vulnerable Web Application Examples from the GHDB

Google Query	Vulnerability Description
intitle:"X7 Chat Help Center" | "Powered By X7 Chat" -milw0rm -exploit	X7 Chat <=2.0 allows remote command execution.
allinurl:tseekdir.cgi	Certain versions of tseekdir.cgi allows local file enumeration.
Copyright. Nucleus CMS v3.22 . Valid XHTML 1.0 Strict. Valid CSS. Back to top -demo -"deadly eyes"	Nucleus 3.22 CMS allows arbitrary remote file inclusion.
"powered by pppblog v 0.3.(.)"	pppblog 0.3.x allows system information disclosure.
"Powered by PHP-Fusion v6.00.110" | "Powered by PHP-Fusion v6.00.2." | "Powered by PHP-Fusion v6.00.3." -v6.00.400 -johnny.ihackstuff	PHP-Fusion 6.00.3 and 6.00.4 contains multiple vulnerabilities.
intitle:"XOOPS Site" intitle:"Just Use it!" | "powered by xoops (2.0)| (2.0.....)"	XOOPS 2.x allows file overwrite.
inurl:wp-login.php +Register Username Password "remember me" -echo -trac -footwear	Wordpress 2.x allows remote command execution.
"powered by ubbthreads"	Certain versions of ubbthreads are vulnerable to file inclusion.
"Powered by sendcard - an advanced PHP e-card program" -site:sendcard.org	Certain versions of Sendcard allow remote command execution.
"powered by xmb"	XMB <=1.9.6 Final allows remote command execution and SQL injection.
"powered by minibb forum software"	Certain versions of minibb forum software allow arbitrary remote file inclusion.
inurl:eStore/index.cgi?	Certain versions of eStore allow directory traversal.[1]

This table and associated GHDB entries provided by many members of the community, listed here by the number of contributions: rgod (85), Joshua Brashars (18), klouw (18), Fr0zen (10), MacUK (8), renegade334 (7), webby_guy (7), CP (6), cybercide (5), jeffball55 (5), JimmyNeutron (5), murfie (4), FiZiX (4), sfd (3), ThePsyko (2), wolveso (2), Deeper (2), HaVoC88 (2), l0om (2), Mac (2), rar (2), GIGO (2), urban (1), demonio (1), ThrowedOff (1), plaztic (1), Vipsta (1), golfo (1), xlockex (1), hevnsnt (1), none90810 (1), hermes (1), blue_matrix (1), Kai (1), good-

virus (1), Ronald MacDonald (1), ujen (1), Demonic_Angel (1), zawa (1), Stealth05 (1), maveric (1), MERLiiN (1), norocosul_alex R00t (1), abinidi (1), Brasileiro (1), ZyMoTiCo (1), TechStep (1), sylex (1), QuadsteR (1), ghooli (1)

Locating Targets Via CGI Scanning

One of the oldest and most familiar techniques for locating vulnerable Web servers is through the use of a *CGI scanner.* These programs parse a list of known "bad" or vulnerable Web files and attempt to locate those files on a Web server. Based on various response codes, the scanner could detect the presence of these potentially vulnerable files. A CGI scanner can list vulnerable files and directories in a data file, such as the snippet shown here:

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

Instead of connecting directly to a target server, an attacker could use Google to locate servers that might be hosting these potentially vulnerable files and directories by converting each line into a Google query. For example, the first line searches for a filename userreg.cgi located in a directory called cgi-bin. Converting this to a Google query is fairly simple in this case, as a search for *inurl:/cgi-bin/userreg.cgi* shows in Figure 6.19.

This search locates many hosts that are running the supposedly vulnerable program. There is certainly no guarantee that the program Google detected is the vulnerable program. This highlights one of the biggest problems with CGI scanner programs. The mere existence of a file or directory does not necessarily indicate that a vulnerability is present. Still, there is no shortage of these types of scanner programs on the Web, each of which provides the potential for many different Google queries.

Figure 6.19 A Single CGI Scan-Style Query

There are other ways to go after CGI-type files. For example, the *filetype* operator can be used to find the actual CGI program, even outside the context of the parent cgi-bin directory, with a query such as *filetype:cgi inurl:userreg.cgi*. This locates more results, but unfortunately, this search is even more sketchy, since the cgi-bin directory is an indicator that the program is in fact a CGI program. Depending on the configuration of the server, the userreg.cgi program might be a text file, not an executable, making exploitation of the program interesting, if not altogether impossible!

Another even sketchier way of finding this file is via a directory listing with a query such as *intitle:index.of userreg.cgi*. This query returns no hits at the time of this writing, and for good reason. Directory listings are not nearly as common as URLs on the Web, and a directory listing containing a file this specific is a rare occurrence indeed.

Underground Googling…

Automated CGI Scanning Via Google

Obviously, automation is required to effectively search Google in this way, but two tools, Wikto (from www.sensepost.com) and Gooscan (from http://Johnny.ihackstuff.com) both perform automated Google and CGI scanning. The Wikto tool uses the Google API; Gooscan does not. See the Protection chapter for more details about these tools.

Summary

There are so many ways to locate exploit code that it's nearly impossible to categorize them all. Google can be used to search the Web for sites that host public exploits, and in some cases you might stumble on "private" sites that host tools as well. Bear in mind that many exploits are not posted to the Web. New (or 0day) exploits are guarded very closely in many circles, and an open public Web page is the *last* place a competent attacker is going to stash his or her tools. If a toolkit is online, it is most likely encrypted or at least password protected to prevent dissemination, which would alert the community, resulting in the eventual lockdown of potential targets. This isn't to say that new, unpublished exploits are *not* online, but frankly it's often easier to build relationships with those in the know. Still, there's nothing wrong with having a nice hit list of public exploit sites, and Google is great at collecting those with simple queries that include the words *exploit, vulnerability,* or *vulnerable*. Google can also be used to locate source code by focusing on certain strings that appear in that type of code.

Locating potential targets with Google is a fairly straightforward process, requiring nothing more than a unique string presented by a vulnerable Web application. In some cases these strings can be culled from demonstration applications that a vendor provides. In other cases, an attacker might need to download the product or source code to locate a string to use in a Google query. Either way, a public Web application exploit announcement, combined with the power of Google, leaves little time for a defender to secure a vulnerable application or server.

Solutions Fast Track

Locating Exploit Code

- ☑ Public exploit sites can be located by focusing on common strings like *exploit* or *vulnerability*. To narrow the results, the *filetype* operator can be added to the query to locate exploits written in a particular programming language.
- ☑ Exploit code can be located by focusing either on the file extension with *filetype* or on strings commonly found in that type of source code, such as *"include <stdio.h>"* for C programs.

Google Code Search

- ☑ Google's Code Search (www.google.com/codesearch) can be used to search inside of program code, but it can also be used to find programming flaws that lead to vulnerabilities.

Locating Malware

- ☑ Google's binary search feature can be used to profile executables, but it can also be used to locate live malware on the web. See H.D. Moore's search engine at http://metasploit.com/research/misc/mwsearch.

Locating Vulnerable Targets

- ☑ Attackers can locate potential targets by focusing on strings presented in a vulnerable application's demonstration installation provided by the software vendor.
- ☑ Attackers can also download and optionally install a vulnerable product to locate specific strings the application displays.
- ☑ Regardless of how a string is obtained, it can easily be converted into a Google query, drastically narrowing the time a defender has to secure a site after a public vulnerability announcement.

Links to Sites

- ☑ www.sensepost.com/research/wikto/ Wikto, an excellent Google and Web scanner.
- ☑ www.cirt.net/code/nikto.shtml Nikto, an excellent Web scanner.
- ☑ http://packetstormsecurity.com/ An excellent site for tools and exploits.
- ☑ Ilia Alshanetsky http://ilia.ws/archives/133-Google-Code-Search-Hackers-best-friend.html
- ☑ Nitesh Dhanjani http://dhanjani.com/archives/2006/10/using_google_code_search_to_fi.html
- ☑ Chris Shiflett http://shiflett.org/blog/2006/oct/google-code-search-for-security-vulnerabilities
- ☑ Stephen de Vries http://www.securityfocus.com/archive/107/447729/30/0

Michael Sutton's Blog:

- ☑ http://portal.spidynamics.com/blogs/msutton/archive/2006/09/26/How-Prevalent-Are-SQL-Injection-Vulnerabilities_3F00_.aspx
- ☑ http://portal.spidynamics.com/blogs/msutton/archive/2007/01/31/How-Prevalent-Are-XSS-Vulnerabilities_3F00_.aspx

- ☑ Jose Nazario's page on Google Code Search insecurity stats: http://monkey.org/~jose/blog/viewpage.php?page=google_code_search_stats
- ☑ Static Code Analysis with Google by Aaron Campbell: http://asert.arbornetworks.com/2006/10/static-code-analysis-using-google-code-search/
- ☑ HD Moore's Malware Search http://metasploit.com/research/misc/mwsearch

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form.

Q: CGI scanning tools have been around for years and have large scan databases with contributions from many hackers. What's the advantage of using Google, which depends on a site having been crawled by Googlebot? Doesn't that give fewer results?

A: Although this is true, Google provides some level of anonymity because it can show the cached pages using the *strip=1* parameter, so the attacker's IP (black or white) is not logged at the server. Check out the Nikto code in Chapter 12, which combines the power of Google with the Nikto database!

Q: Are there any generic techniques for locating known vulnerable Web applications?

A: Try combining *INURL:["parameter="]* with *FILETYPE:[ext]* and *INURL:[scriptname]* using information from the security advisory. In some cases, version information might not always appear on the target's page. If you're searching for version information, remember that each digit counts as a word, so 1.4.2 is three words according to Google. You could hit the search word limit fast.

Also remember that for Google to show a result, the site must have been crawled earlier. If that's not the case, try using a more generic search such as *"powered by XYZ"* to locate pages that could be running a particular family of software.