Global Information Assurance Certification Paper

Basics of CGI Security
Common Gateway Interface, CGI, at a Glance

By Jeffrey McKay

Introduction

In these times of the Internet explosion a company is hard pressed to do business in the electronic world without an enticing web page. To keep a potential customer's attention, web pages should download to the users screen in less than 10 seconds or that user will most likely move on. The same thing can be said for the look and feel of a web page as well as its functionality, content and search capabilities. If you've used a Web search engine to find an item of interest, you've used the Common Gateway Interface or CGI. But CGI itself isn't the search engine; however, it's a way to interface programs, such as search engines, with Web servers.[1] For a user to request a specific product or search topic, it is usually input into a "web form" and sent to the web server via a GET or a POST command. This data is sometimes sent to a Common Gateway Interface or CGI script program and acted upon accordingly.

This is where the problems start. When data is entered into an online form it can sometimes be manipulated to take advantage of certain holes in existing code, default configurations and/or files. Often times, the LAN administrator, with no web development experience, is called upon to bring up the company's first web server and these holes are not plugged. This is usually the first place hackers often look to gain entry. In 1999 alone there were 31 cgi-bin attacks documented in Mitre's Common Vulnerabilities and Exposures (CVE). In 2000 there were 46 CGI attacks, a 50% increase. These figures do not even include the "Candidate" or under review attacks that are still being verified from those years.[2]

This paper will describe some of the basic pitfalls of CGI programming and web development. It will hopefully alert you to the initial issues that must be resolved in order to protect your site and company reputation. It will touch upon what CGI is, the common cgi vulnerabilities, web hacking tools, and problem correction/installation issues.

What is CGI?

Early web development depended upon Hyper Text Markup Language, HTML, as the language to statically define web pages. These were fixed, read only documents that could be referenced by mouse movement but not queried. Again, if you've used a Web search engine to find an item of interest, you've used CGI. But CGI itself isn't the search engine. It's a way to interface programs, such as search engines, with Web servers. HTTP (Web) servers are designed primarily to serve up HTML documents. But CGI files are not documents they are programs. Therefore, to store CGI programs most Web

servers use a special directory, commonly named cgi-bin. The Web server knows that files stored in the cgi-bin directory are to be executed rather than simply sent to the user's Web browser. CGI programs can be written in a wide variety of languages, including DOS batch files, BASIC, C, and scripting languages such as Perl. It's the job of the CGI to activate the CGI program at the proper time and pass the program to any necessary user or operating environment-generated data. The CGI program then processes whatever data is generated. Once the program accomplishes this process, it must return some output to the user via the user's Web browser. When programs are generated from user input it is an avenue of entry for the hacker.[1] The hacker can make use of poorly written code to put more data into a field than allowed for instance. This causes a buffer-over-run that can run hacking code into special areas on the web server.

Common CGI Vulnerabilities

For several years Mitre (http://cve.mitre.org) has managed a "database" or what they call a "dictionary" of standardized names of vulnerabilities and security exposure information. Their goal is to take all publicly known vulnerabilities and make it available for users to search for this information in other databases. They state that the "CVE should not be considered as a vulnerability database on its own merit."[2]

When searching for "cgi" in Mitre's dictionary, most of the vulnerabilities were localized to remote execution of programs or viewing of files. Programs on the web servers allowed, for example, via the cgi.phf program, remote command execution through shell metacharacters. Metacharacters, "../" for instance, allow a change of "back one directory." Metacharacters can include:

/ \ " ' " ~ ; [] {} | * & ^ % $ # !

Other vulnerabilities allowed shell access from a buffer overflow, i.e. in the php.cgi program. For any web developer this page should be bookmarked and it should be a ritual to look for updates.

The following are a few candidates from the CGI Common Vulnerabilities and Exposures from Mitre. Candidate vulnerabilities must be reviewed and accepted by the CVE Editorial Board before they can be added to the official CVE list. Therefore, these candidates may be modified or even rejected in the future. They are provided for use by individuals who have a need for an early numbering scheme for items that have not been fully reviewed by the Editorial Board. They are nonetheless very interesting.

> CVE-1999-0174 The view-source CGI program allows remote attackers to read arbitrary files via a .. (dot dot) attack.
> CVE-1999-0191 IIS newdsn.exe CGI script allows remote users to overwrite files.
> CVE-1999-0236 ScriptAlias directory in NCSA and Apache httpd allowed attackers to read CGI programs.
> CVE-1999-0237 Remote execution of arbitrary commands through Guestbook CGI program.

CVE-1999-0260 The jj CGI program allows command execution via shell metacharacters.

CAN-2001-0211 ** CANDIDATE (under review) ** Directory traversal vulnerability in WebSPIRS 3.1 allows remote attackers to read arbitrary files via a .. (dot dot) attack on the sp.nextform parameter.
CAN-2001-0214 ** CANDIDATE (under review) ** Way-board CGI program allows remote attackers to read arbitrary files by specifying the filename in the db parameter and terminating the filename with a null byte.
CAN-2001-0224 ** CANDIDATE (under review) ** Muscat Empower CGI program allows remote attackers to obtain the absolute pathname of the server via an invalid request in the DB parameter.
CAN-2001-0231 ** CANDIDATE (under review) ** Directory traversal vulnerability in newsdesk.cgi in News Desk 1.2 allows remote attackers to read arbitrary files via a .. in the "t" parameter.
CAN-2001-0232 ** CANDIDATE (under review) ** newsdesk.cgi in News Desk 1.2 allows remote attackers to read arbitrary files via shell metacharacters.[2]
SNIP

You can see that some of the same vulnerabilities from 1999 are still showing up in 2001 in slightly different forms. It all bears watching and hardening your web server as noted later.

CGI Hacking Tools

How do hackers find their way into your web server so easily? The security community itself, for the most part, writes programs to test their own systems and then release it to the community as a simple program that a 10 year old could run. Often times, someone finds a vulnerability into a system by chance and then, mostly for notoriety, quickly releases it to the general public. Then of course there is a rash of break-ins at various companies. The security community generally frowns upon this, as the companies are not given sufficient time to correct their code. The user community is also often slow to install these updates thus compounding the problems.

One of the most robust CGI scanning programs is Whisker, written by Rain Forest Puppy (rfp). RFP himself has said this is a "next generation CGI scanner" and is "not lame" as one would think. It is a scriptable "programming-ish" language that can be tailored to evade intrusion detection systems. It also has the logic to avoid scanning for data that would never be there in the first place thus tipping off an Intrusion Detection System (IDS). For instance if you are scanning an Apache server it will not check for .htr handlers that aren't there in the first place. The IDS system may or may not be watching for this. If a customer does use apache as their web server and they get one .htr scan and the IDS is set up to watch for this, Whisker would avoid tripping it with a "stupid scan."

It has a database of over 90 configurable web server types to avoid being detected this way, but most importantly has over 130+ sample scripts to utilize for scanning. It even has options for reading nmap output, files full of domains, single and virtual hosts. [3]

The output from Whisker is impressive:

-- whisker / v1.4.0 / rain forest puppy / www.wiretrip.net --
- Loaded script database of 1968 lines

= - = - = - = - = - =
= Host: a.10.1.1
= Server: Apache/1.3.6 (Unix) PHP/4.0.3pl1 mod_ssl/2.3.6 OpenSSL/0.9.3a

+ 302 Found: GET /cfdocs/
+ 302 Found: GET /scripts/
+ 302 Found: GET /cfcache.map
+ 302 Found: GET /cfide/Administrator/startstop.html
+ 302 Found: GET /cfappman/index.cfm
+ 403 Forbidden: GET /cgi-bin/
+ 404 Not Found: GET /cgi-bin/dbmlparser.exe
+ 302 Found: HEAD /_vti_inf.html
+ 302 Found: HEAD /_vti_pvt/
+ 404 Not Found: HEAD /cgi-bin/webdist.cgi
+ 404 Not Found: HEAD /cgi-bin/handler
+ 404 Not Found: HEAD /cgi-bin/wrap
+ 404 Not Found: HEAD /cgi-bin/pfdisplay.cgi
+ 404 Not Found: HEAD /cgi-bin/MachineInfo
+ 302 Found: HEAD /PDG_Cart/
+ 302 Found: HEAD /quikstore.cfg
+ 302 Found: HEAD /orders/
+ 302 Found: HEAD /Admin_files/order.log
+ 302 Found: HEAD /WebShop/
+ 302 Found: HEAD /pw/storemgr.pw
+ 302 Found: HEAD /bigconf.cgi
+ 302 Found: HEAD /icat
+ 200 OK: HEAD /Fpadmin

As you can see there are many things Whisker can tell you about a sites layout. Other CGI scanning programs, like wCGIchk[4], are similar to Whisker. There are many other web pilfering programs including some very powerful commercial programs. (See http://www.hackingexposed.com) For instance, Teleport Pro for NT can mirror an entire site on your system for further review. It can even set search criteria to look for certain keywords. One way to prevent or track for such a program is to monitor your logs for the fast GET requests this program would generate from a single user. You can also make a

honeypot type CGI script to produce an endless stream of garbage to this type of user.[5]

Correcting the Problem

There are many ways to prevent cgi attacks and the easiest and most sensible is to install your web server correctly from the start. Many web servers have insecure CGI programs that are actually supplied with the default installation program. For instance the PHF Phone Book Script is a very old vulnerability and is really not used anymore yet it is a standard file on some of today's web server installations. This file should be removed immediately as it not used anymore.

Vulnerabilities in the web program should be checked with Mitre.org, CERT (www.cert.org) and Bugtraq (www.securityfocus.com) sites and the web server's manufacturer for patches and updates. The operating system itself must of course be brought up to date with the latest patches. All web CGI programs must be checked for insecurities via scanners to see if they are tempting targets for hackers. As previously mentioned, web developers often write programs and have little regard for the security of their work. They are forced to produce a workable product as fast as possible and security concerns fall by the wayside.
These basic tenets should be followed and time and consideration given to the developers for a workable and secure web program.
If your CGI programs create or open files, observe these rules:

- Always include error-handling code to warn you if the file isn't actually a file, cannot be created or opened, already exists, doesn't exist, requires different permissions, and so on.
- Watch which directories you use to create or open files. Never write a file to a world-writeable or world-readable directory.
- Always explicitly set the file's UMASK.
- Set the file permissions as restrictively as possible. If the file is a dump of user input, such as a visitor list, the file should be readable only by the processes that will engage that file.
- Ensure that the file's name does not have metacharacters in it, and if the file is generated on the fly, include a screening process to weed out such characters.[6]
- Never run your web server with root privileges.
- Delete scripts that are not in use to prevent a vulnerability from being exploited.
- Deploy Intrusion Detection Systems to alert you to snooping on your network.
- Deploy Host Intrusion Detection programs that alert on changes to files, processes that are run or not run, Registry or Kernel changes and common vulnerabilities.
- Protect the logs of your web server. Store logs on a secure system that can alert you to any changes.

- Utilize CGIWrap written by Nathan Neulinger (<nneul@umr.edu>) to allow general users to use CGI scripts and HTML forms without compromising the security of the http server. Scripts should be run with the permissions of the user who own the script, rather than using the user ID of the httpd process. Also, several security checks are performed on the script, which will not be executed if any checks fail.
- If you download a CGI script from the web defiantly check it for flaws or hacks:
    1. How complex is it? The longer it is, the more likely it is to have problems.
    2. Does it read or write files on the host system? Programs that read files may inadvertently violate access restrictions you've set up, or pass sensitive system information to hackers. Programs that write files have the potential to modify or damage documents, or, in the worst case, introduce Trojan horses to your system.
    3. Does it interact with other programs on your system? For example, many CGI scripts send e-mail in response to a form input by opening up a connection with the sendmail program. Is it doing this in a safe way?
    4. Does it run with suid (set-user-id) privileges? In general this is a very dangerous thing and scripts need to have excellent reasons for doing this.
    5. Does the author validate user input from forms? Checking form input is a sign that the author is thinking about security issues.
    6. Does the author use explicit path names when invoking external programs? Relying on the PATH environment variable to resolve partial path names is a dangerous practice.[7]
- Do not give out information regarding your site. Turn finger off!
- If you are coding in C language you must be aware of buffer overflows. Never assume the user input provided by a POST request will never exceed the size of the static input buffer. A hacker can take advantage of this and push more data into the data cell and thus crash the program.[7]
- Do not pass unchecked remote user input to a shell command. This type of input can invoke a /bin/sh subshell. Examples of risky C commands are the popen(), exec() and in Perl the system(), exec(), piped open and eval() functions, all of which can start this process.[7]
- Backticks and shells do not mix either. Backtick quotes, available in shell interpreters and Perl for capturing the output of programs as text strings, are also dangerous.[6]
- Generalities:
    1. If your company runs its own web page or even if you utilize a web hosting company, you should have a security policy in place. Also there should be some kind of security routine or awareness.
    2. Education is key in keeping up with security issues. It gives you the

foundation to build upon for the daily changes in the web server environment.

3. Know the intricacies of your web environment. If you have Apache web servers or certain application programming interfaces, you should utilize all available resources i.e. web pages and security mailing lists. A great web security resource is: http://www.lanw.com/training/interop/securityurls.htm.[8]

Conclusion

In conclusion, there are many ways to protect your web site from CGI attacks. It makes perfect sense to set up the web site correctly from the start but often times work deadlines overrule sensible secure programming. The tips above are only a meant to be a preliminary input to your new web server Security policy. The links referenced are a valuable source for further research into web server CGI security.

---

[1] Frank, Alan. HTML and CGI, Part One. 1 August 1996. http://www.networkmagazine.com/article/NMG20000721S0008

[2] Mitre.org, Common Vulnerabilities and Exposures. April 4,2001 http://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=cgi

[3] Whisker CGI Scanner supported by Rain Forest Puppy. April 4,2001 http://www.wiretrip.n et/

[4] Maximum Linux Security, A Hacker's Guide to Protecting Your Linux Server and Workstation, Anonymou s

[5] CGIWrap Utility. April 4 2001 http://sourceforge.net/projects/cgiwra p/

[6] "Hacking Exposed," by Stuart McClure, Joel Scambray and George Kurtz, Osborne/McGraw-Hill, Berkeley, Calif., 1999, ISBN: 0-07021212 7-0

[7] The World Wide Web Security FAQs. April 2 2000 http://www.w3.org/Security/faq/wwwsf4.ht ml

[8] Web site security does matter. Ed Tittel 18 Aug 2000

http://searchsecurity.techtarget.com/tip/1,289483,sid14_gci297744,0