Detecting CGI script abuse

http://advosys.ca/tips/cgi−trap.html
Jul 15 2000

This web tip shows one way to detect crackers probing your web server's CGI−BIN for security holes.

Background

Most Internet servers sit behind firewalls and use detection scripts to send alerts when break−ins are attempted. Some system administrators even run software to detect portscanners and denial−of−service attempts. However, many system administrators still overlook security problems in CGI scripts and web applications.

As demonstrated by recent security alerts, improperly written CGI scripts and web applications can let crackers read system files, obtain passwords, crash the server or worse. A system may be firewalled and hardened against remote logins, FTP access and denial of service attacks, yet have many well−known holes in the server's web applications and CGI scripts.

Beyond the difficulty of writing secure web applications that all web developers face, many web servers and application servers ship with demonstration scripts that have severe security vulnerabilities. For example:

- Microsoft IIS 4.0 example ASP script holes
- Allaire Cold Fusion 3.0 and 4.0 sample script holes
- NCSA and Apache httpd Phf script vulnerability

The PHF script listed above has not been included with Apache for several years, but it became so widely known that crackers are still scanning web server cgi−bin directories for it. Lists of well−known CGI script holes are available on most cracker web sites along with scripts that scan for each one.

Diligent system administrators know about these vulnerabilities and have removed dangerous scripts. However it would still be good to know when someone comes looking for them on your web server. Scanning for CGI and web application holes is usually only part of a cracker's attempt to inventory a server. Knowing when you are being probed for CGI holes can signal you to look for other crack attempts.

Trapping 404 errors:

One method we have used to detect CGI script abuse is to redirect all "404 Not Found" responses from the web server to a script that examines the request for suspicious activity.

Both Apache and iPlanet / Netscape web servers allow customized error messages. Either custom HTML files can be output or scripts can be executed in response to any of the standard HTTP request errors. Since a properly secured web server will not contain any of the well−known CGI vulnerabilities, any attempt by an outside to look for them results in a 404 Not Found response.

Apache comes with an example script "phf_abuse_log.cgi" you can use to log attempt to access phf. This concept can be expanded to look for any suspicious URL request (such any request containing "/etc/passwd" on a unix server) each time a 404 error is raised in the CGI−BIN directory. Instead of simply writing the activity to a file, it would be better to immediately e−mail the system administrator or trigger the server's

monitoring software.

A detection script

We have written an example CGI script in Perl to be installed as a "404 Not found" error handler in Apache or Netscape Enterprise Server. It detects attempts to access well−known vulnerable CGI scripts, as well as any attempt containing the strings "etc/passwd" and "etc/shadow" (the user database files on Unix servers).

When a suspicious HTTP request is received, the script calls Sendmail to e−mail the system administrator. The mail alert contains date, time, remote IP address of the attacker and URI attempted.

Access to the following well−known vulnerable CGI scripts are detected:

- phf
- jj
- campas
- htmlscript
- test−cgi
- websendmail
- count.cgi

The script outputs a normal "404 Not found" message unless suspicious activity is found. If abuse is detected, a "403 Forbidden" message is returned with the simple statement "Your attempt as been logged".

```
#!/usr/bin/perl
#
# abuse_trap
#
# ErrorDocument handler for Apache and Netscape Enterprise
# to examine all "not found" HTTP requests for CGI abuse attempts.
#
# Written by Advosys Consulting Inc., Ottawa
# Based on phf_abuse_log.cgi in the Apache 1.3.3 source code distribution.
#
# Provided AS−IS without warranty of any kind.
#
# Revision history:
#  Nov 24 1998: Version 1.0
#
# Install this script as your web server's '404 Not Found' error handler.
#
# Apache: Insert the following lines into httpd.conf:
#           ScriptAlias /abuse_trap /opt/apache/cgi−bin/abuse_trap
#           ErrorDocument 404 /abuse_trap
# iPlanet: In the server Admin, go to Server Preferences −> Error Responses.
#          In the "Error code not found" field put <path−to−cgi−bin>/abuse_trap
#          as the file and check the "CGI" checkbox.


# E−mail address to send alerts to
$MAILTO = 'security@yourdomain.com';

# Regexp of CGI request strings to watch for:
$forbidden = 'etc\/passwd|etc\/shadow|phf|jj|campas|htmlscript|test−cgi|websendmail|count.cgi';

### Main routine begins here
###
```

```
require "ctime.pl";


# iPlanet and Apache use different variables (of course):
if ( $ENV{SERVER_SOFTWARE} =~ /apache/i ) {
        $request = $ENV{REQUEST_URI}
}
else {
        $request = $ENV{PATH_INFO};
        $request .= "?" . $ENV{QUERY_STRING} if $ENV{QUERY_STRING};
}

if ( $request =~ /\/cgi/  and  $request =~ /$forbidden/i ) {
        $pagestatus = '403 Forbidden';
        $message = 'Your attempt has been logged.';
        print_html_message();
        mail_warning();
}
else {
        $pagestatus = '404 Not found';
        $message = 'The requested object does not exist on this server. The link you followed is
        print_html_message();
}

sub print_html_message {
print "Content-type: text/html\n";
print "Status: $pagestatus\n";
print "\n";
print "<BODY>\n";
print "<H1>$pagestatus</H1><P>\n";
print $message;
print "</BODY>";
}

sub mail_warning() {
# Use sendmail to send a warning message
$when =
$when =~ s/\n//go;
$ENV{HTTP_USER_AGENT} .= " via $ENV{HTTP_VIA}" if($ENV{HTTP_VIA});
$HOSTNAME = `uname -n`;
$HOSTNAME =~ s/\n//go;

open (MAIL,"| /usr/lib/sendmail -t -oi") or die;
print MAIL <<END;
To: $MAILTO
From: CGI abuse logger (root\@$HOSTNAME)
Subject: CGI abuse attempt on $ENV{HTTP_HOST} ($HOSTNAME)

An attempt to access a CGI hole was detected on site $ENV{HTTP_HOST} on $HOSTNAME

Please see the web site's access logs for further details.

Time:           [$when]
Web site name:  $ENV{HTTP_HOST}
Server name:    $HOSTNAME
Web site admin: $ENV{SERVER_ADMIN}

URI attempted:  $request
Remote address: $ENV{REMOTE_ADDR}
Remote host:    $ENV{REMOTE_HOST}
HTTP referrer:  $ENV{HTTP_REFERER}
Web Browser:    $ENV{HTTP_USER_AGENT}
```

```
(This message was automatically generated by the ErrorDocument script running under $ENV{SERVER_
END

close MAIL;
}
```

Note:

- Change the e−mail address for the $MAILTO variable to the address you want alerts to go to (line 27)
- Point the script to the location of Sendmail on your server if it is not /usr/lib/sendmail (line 77)

The script has been tested on Linux and Sun Solaris 2.5 with Perl 5.004. It could probably be ported to MS Windows without too much trouble.

Configuring Apache

To configure apache to run the abuse trap script each time a "404 Not Found" error occurs in the cgi−bin directory, add directives to your httpd.conf or access.conf similar to the following:

```
ScriptAlias /abuse_trap /usr/lib/cgi−bin/abuse_trap.pl
<Directory /usr/local/cgi−bin>
ErrorDocument 404 /abuse_trap
</Directory>
```

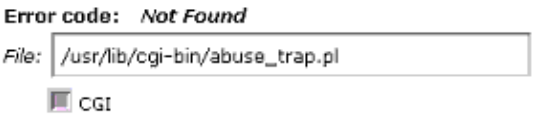
Substitute the location of your web server's CGI−BIN directory in the above if it is not /usr/local/cgi−bin

Configuring iPlanet / Netscape Enterprise

To configure iPlanet Enterprise Web Server (formerly called Netscape Enterprise Server), access the iPlanet Administration Server and select the server instance you wish to manage. Select the "Server Preferences" tab then "Error Responses" from the sub−menu.

In the blank labelled "Error code: Not Found" enter the full path and name of the abuse trap script and check the "CGI" checkbox.

For example:



Don't forget to save and apply the changes.

Testing the script

Once intalled and your Apache or iPlanet server have been restarted, test the script from a web browser: attempting to access a suspicous URL such as "http://your.webserver.com/cgi−bin/badscript?/etc/passwd"

Mail should be sent to the address you specified in the abuse trap CGI script with a format similar to the following:

```
Date: Sat, 4 Sep 1999 20:42:23 −0400
To: security@yourdomain.com
From: CGI.abuse.logger@mywebsite.com
Subject: CGI abuse attempt on www.mywebsite.com

An attempt to access a CGI hole was detected on
site www.mywebsite.com on host dustpuppy

Please see the web site's access logs for further details.

Time:           [Sat Sep  4 5:40:00 1999]
Web site name:  www.mywebsite.com
Server name:    dustpuppy
Web site admin: webmaster@mywebsite.com

URI attempted:  /cgi/badscript?etc/passwd
Remote address: 192.168.1.5
Remote host:    nasty.evilcracker.org
HTTP referrer:
Web Browser:    Mozilla/4.61 [en] (Win98; U)

(This message was automatically generated by the
ErrorDocument script running under Apache/1.3.9
(Unix) on host dustpuppy.)
```

Enhancements

This abuse trap script is functional, but is really just a starting point for your own custom version.

You can extend the script to scan for attempts to access the newer IIS and Cold Fusion sample script vulnerabilities or other important system files. The alert action can be changed to trigger a network manager system alarm, automatically firewall off the attacking IP, or any number of responses.

Resist the urge to add threats and insults to the user message. The script performs simple pattern matching and could be possibly set off by a legitimate user accessing a misspelled URL. In the case of an real attack, do you really want to motivate a cracker into crashing your server by insulting or challenging them?

As the message sent by the script suggests, alerts generated by the script should be verified by examining the web server logs. Other system logs should be scanned by the administrator for other attempts to break into the system... CGI probes usually come as part of an overall attempt to inventory a server for vulnerabilities of any kind.

More information

Checking "404 Not Found" errors for suspicious activity will not catch all attempts to exploit CGI script or web application holes.

There is no substitute for due diligence by developers to create secure code in the first place. System administrators must also delete sample scripts and configuration holes when installing web and application server software on production Internet servers. Everyone is responsible for staying up−to−date on known vulnerabilities.

Many organizations track and report on web server security problems, including CGI script holes. Here are a few recommended sites:

- Computer Emergency Response Team (CERT) (USA)
- l0pht Heavy Industries
- Rootshell

Also, check out our web tip Writing secure web applications for suggestions on good programming practices for web application development.

Comments, suggestions, criticisms, additions to this document?
Please e−mail tips@advosys.ca

Copyright and terms of use

Use of this document

Permission to use this document from the Advosys Consulting web site is granted, provided that (1) This notice notice appears in all copies, (2) use is for informational and non−commercial or personal use only and will not be copied, reprinted, or posted on any network, computer or broadcast in any media, and (3) no modifications of the document are made.

Educational institutions (specifically K−12, universities and community colleges) may reproduce the Documents for distribution in the classroom, provided that (1) the below copyright notice appears on all copies, and (2) the original Uniform Resource Locator ("URL") of the document on the Advosys Consulting web site appears on all copies.

Use of this document for any other purpose requires written permission of Advosys Consulting Inc.

Limitation of liability

Advosys Consulting take no responsibility for the accuracy or validity of any claims or statements contained in the Documents and related graphics ("the content") on the Advosys web site. Further, Advosys Consulting Inc. makes no representations about the suitability of any of the information contained in the content for any purpose. All such documents, related graphics, products and services are provided "as is" and without warranties or conditions of any kind. In no event shall Advosys Consulting Inc. be liable for any damages whatsoever, including special, indirect or consequential damages, arising out of or in connection with the use or performance of information, products or services available on or through the Advosys Site.

Trademarks

Product, brand and company names and logos used on the Advosys web site site are the property of their respective owners.

About Advosys Consulting

Advosys Consulting is a privately held systems management corporation. Our global headquarters is in Ottawa, Ontario Canada. Formerly known as "Webber Technical Services", we have been providing systems management, computer engineering, security and consulting to private sector and government clients since 1991.

Areas of expertise

Advosys is a diversified consulting firm providing services in many areas of Information technology:

- Internet technologies
- Firewalls and information security
- Web applications
- Network architecture
- Unix and Linux systems management

Unbiased recommendations

Advosys Consulting is an *independent* consulting firm. We have broad experience with multiple vendors including Sun, Hewlett−Packard and Microsoft but are not a reseller of their hardware or software.

Unlike many consulting firms, Advosys receives no commissions, percentages, or other rewards from companies for promoting particular products or services.

This allows us the freedom to offer uncompromised objectivity. We have knowledge and experience with a broad range of products and technologies and can recommend solutions from any manufacturer, or open−source freeware if that is what best fits the requirements. Advosys works only for you, our client, not for a product manufacturer.

For more information, please visit us at http://advosys.ca