GoAhead WebServer 2.5 Release Notes

Enhancements

PeerSec Networks MatrixSSL Layer

Change

MatrixSSL 3 embedded SSL is now easily integrated in the WebServer 2.5 package. Several files were modified/added, and Makefiles were changed to accommodate the change.

Result

It is easier than ever to include SSL/TLS support in WebServer 2.5. Simply unpack the latest MatrixSSL version to the root Webs 2.5 directory and compile. SSL allows WebServer to process https:// URLs. Integrated with the user management features of WebServer, secure sessions can now be handled by the server without additional software. An additional API, websRequireSSL() is defined in websSSL.h to specify URL directories with SSL access only.

Whitelist-Based File Access

Change

WebServer now uses a whitelist-based approach to determine file accessibility. The Web server will periodically scan the websDefaultDir (www root) recursively looking for files and add them to the whitelist. When a file is requested from the Web server, the Web server will only allow files that have been found via this scan to be accessed.

Result

Numerous bugs had been reported across numerous platforms regarding the ability to access files outside the web root directory by crafting special URLs that had unintended consequences. The addition of whitelist validation standardizes the URL-to-file lookup across all platforms to a very basic case-insensitive string comparison, greatly improving the security and stability of WebServer.

Logging Format Updated

Change

Enabling logging by defining WEBS_LOG_SUPPORT in webs.h now produces a standardized log output in the Apache Common Log Format.

Result

Standardized logs can now be processed with third-party log analysis packages.

Documentation Interface Updated

Change

The documentation interface no longer relies on Java. The new system uses an iframe and small amount of JavaScript to improve compatibility across platforms. Also, the look and feel is updated.

Result

Various systems had incompatibilities with the Java applet, and now they will function properly.

Bug Fixes

uClinux Makefile problem

Issue

The Makefile for uClinux did not correctly specify the compiler name. This is a problem with uClinux builds when doing cross-compilation because the wrong compiler will be used.

Fix

Use Makefile variable $(CC) instead of cc. Credit: PeerSec Networks.

OS X 10.5 Compile Fix

Issue

The included Makefile was not updated for Mac OS X 10.5.

Fix

Updated Makefile to compile properly on OS X. Credit: PeerSec Networks.

Incorrectly returning 200 status code for resource not found

Issue

When forms, asp pages, or cgi scrips were not found for the given URL, an HTTP 200 status was returned, providing an explanatory message. This is not in line with the HTTP specification which indicates a 404 error should be returned for resource not found.

Fix

Return 404 instead of 200 in form.c, asp.c, and cgi.c. Credit: PeerSec Networks.

Infinite Loop

Issue

Sending the Web server an invalid request that contains no newline character (\n) causes the Web server to get into an infinite loop and to consume 100% CPU usage (at least on Windows).

Removing the #ifdef HP_FIX will terminate the session as soon as there is an error (no data available). This would solve the problem. If data arrival stops, the session is ended. If there is a small delay between two packets in the middle of a request, it will discard the request. This might discard some longer POST or PUT requests on a slow connection.

It's the assumption in socketInputBuffered that if there is data in the lineBuf, the socket is marked as readable in socketSelect() line 822. This is not a correct assumption because the lineBuf is never filled with more than one line at a time, and it is immediately cleared when the line is returned. There will never be a full line waiting in the lineBuf.

Fix

Change in sock.c:365 socketInputBuffered(). Credit: Simon Byholm.

Bad POST causes segfault

Issue

If you set Content-Length to 0 in a POST invoking formTest(), WebServer will GP fault (segfault).

Fix

The problem was in webs.c:websReadEvent(webs_t wp):. Credit: Fabien R.

Segfault in SSL builds

Issue

The problem is caused by websSSLGets not null terminating strings with len=256 (BUF_BLOCK len long). This leads to bQhead corruption and later SEGV.

Fix

Updated websSSL.c and integrated MatrixSSL. Credit: Joakim Tjernlund and PeerSec Networks.

Incorrect fd_mask size

Issue

Affects the eCos platform but noticed on other operating systems and architectures (64-bit builds of HP-UX and Solaris). Assumes sizeof (int) == sizeof(fd_mask).

Fix

The solution was to modify sockGen.c: socketSelect(). Credit: Petchesi Gabriel Horatiu.

Java applet in treeapp.asp not working

Issue

The Java-based navigation panel used to navigate the documentation will not work (although it loads properly) if the pages are in ROM because the applet is using an incorrect URI to access the documentation (without "/"). This is a

problem because if the online documentation is contained in the Web site, you can't navigate properly. This fix is only needed if the documentation is accessed from webrom,.

Fix

The solution was to remove the Java applet and replace it with a more cross-platform UI. Credit: PeerSec Networks.

Java applet does not run

Issue

The Java applet shipped with the GoAhead documentation and loaded in treeapp.asp may not work correctly in some situations. The problem is caused by the security constraints imposed to unsigned Java applets by the JVM. The problem is with the SERVER_ADDR value used as a parameter for the Java applet. This value stays the same regardless of the network interface (IP address) the client connection is coming from. The Web server will bind to all network interfaces available--you can check this with the netstat -ntl command. If a new network interface is brought up after the Web server was started, the Web server will still reply to requests coming trough that interface.

Fix

Integrated a patch to webs.c and webs.h to solve the connection problem. This patch sets SERVER_ADDR as the IP of the interface that accepts the connection. Credit: Carlitos.

Base64 fix

Issue

base64.c has an error in its map64[] lookup table.

Fix

base64.c: the entries for the '+' and '/' symbol are in the wrong place and will erroneously encode the NULL character. Credit: Ian Bothwell.

Can't upload binary files (CGI)

Issue

When implementing file upload capability in external CGI script, uploading binary files does not work, but uploading text files works.

Fix

webs.c: replace gwrite(fd, text, gstrlen(text)); with gwrite(fd, text, nbytes); Strlen should not be used to determine binary lengths. Credit: Holger Lindeberg Bille.

64-bit compatibility

Issue

The WebServer code is not 64-bit ready. It makes the wrong assumption in many places that an int can hold a void *. This is not always the case, so you will encounter problems on any platform where sizeof(int) < sizeof(void *). This could include some embedded environments (no user reports on this). This problem arose for 64-bit builds on the following platforms: HP-UX 11.X, 11.23, Solaris 8,9, and DEC Alpha.

Fix

Updated many files and assumptions where int and void* were used interchangeably: asp.c, ejlntrn,h, ejparse.c, sock.c uemf.h, webs.c, websSSL.c, and websuemf.c. Replaced md5c.c with PeerSec version. Credit: Barry Stone, Petchesi Gabriel Horatiu, and PeerSec Networks.

Can’t upload binary files (CGI)

Issue

ASP files with write() statements containing output of more than 128 characters will cause the GoAhead Web server to leak memory. Repeated calls cause the memory usage of the Web server to keep increasing. Splitting the single write statement into multiple shorter statements is a workaround.

Fix

Fixed in ejlex.c, and inputPutback(). Credit: Fred Sauer.

DAA URI digest

Issue

The Web server calculates the digest using the URI which does not include the request parameters (when using HTTP GET). All the browsers tested (except Safari on Mac OS X) include the parameters in the digest. This is in line with the specification.

Fix

Fixed in security.c. First the digest is calculated without the request URL, and if that fails, it is calculated with the URL so that method can also be valid. Credit: Richard Laing.

Worms (long URL strings)

Issue

There are reports that worms targeting IIS and Apache using long URLs crash the Web server. The attackers use a very long string in the URL that starts something like this:

SEARCH /\220\002\261\002\061 ....

…and goes on forever. The Web server at some point stops responding. Users reported these problems with version 2.1.5 and 2.1.6 running on VxWorks.

Fix
Fixed in socketGets. Maximum input line set to WEBS_MAX_URL (default 2048). Credit: Simon Byholm.

VxWorks does not support O_APPEND flag

Issue
VxWorks does not support O_APPEND flag to open a file; thus, you must seek each time to the end of the stream.

Fix
Changed multiple occurrences of:

fd = gopen(wp->cgiStdin, O_CREAT | O_WRONLY | O_BINARY | O_APPEND, 0666);

to:

fd = gopen(wp->cgiStdin, O_CREAT | O_WRONLY | O_BINARY, 0666);

lseek(fd, 0, SEEK_END);

Credit: newzy at hotmail dot com.

Resource Consumption

Issue
http://aluigi.altervista.org/adv/goahead-adv1.txt

The Web server is affected by a bug that lets an attacker consume all its resources. The attacker uses the POST method with a specific number in the Content-Length parameter (the value that specifies how many bytes will be sent to the server) and then sends an amount of data less than the amount specified.

The server then allocates all the data sent by the attacker and then waits for the last bytes as specified by Content-Length. Then the attacker breaks the connection and the server enters in an infinite loop because the socket's error is not well managed. The following is an example of what the attacker needs to send to the server:

POST / HTTP/1.0

Content-Length: 10

123456789

So the socket will not be closed by the server, all the memory allocated until that moment will not be freed, and the CPU will go to 100% due to the infinite loop of the unchecked select() function. On some operating systems the Web server will accept no more connections after some attacks.

Fix
Fixed in webs.c:websGetInput():587. Explicitly test for socket EOF. http://aluigi.org/patches/goahead-webpostmem-fix.txt. Credit: Luigi Auriemma and Dhanwa T.

Bypassing Directory Traversal Rules

Issue
http://aluigi.altervista.org/adv/goahead-adv2.txt

WebServer has an internal problem that lets it also accept HTTP requests that don't start with the slash or that contain backslashes (both \ and %5c) after or at the same place of the initial slash. So while a correct request, such as "GET /file HTTP/1.0", is accepted, a bad request, such as "GET file HTTP/1.0" or "GET \file HTTP/1.0", or "GET /\\%5cfile HTTP/1.0", is also accepted. This bug leads to the system bypassing management of the "special" directories (as cgi-bin) and to their use as normal "unmanaged" directories. In fact, the server uses a function called websUrlHandlerDefine() letting the admin specify how to manage each directory. For example, by default we have

```
websUrlHandlerDefine(T("/goform"), NULL, 0, websFormHandler, 0);
```

used to execute the built-in functions written by the same admin or

```
websUrlHandlerDefine(T("/cgi-bin"), NULL, 0, websCgiHandler, 0);
```

used just to manage the cgi-bin directory. So, if an attacker uses a bad HTTP request, the attacker will easily bypass the management decided by the admin for a specific directory (function websUrlHandlerRequest()). The most common and useful effect of this bug is the download and the viewing of any file in the cgi-bin directory. For example:

http://server/\cgi-bin/cgitest.c	"GET cgi-bin/cgitest.c HTTP/1.0"
http://server/\\\cgi-bin/cgitest.c	"GET \cgi-bin/cgitest.c HTTP/1.0"
http://server/%5ccgi-bin/cgitest.c	"GET %5ccgi-bin/cgitest.c HTTP/1.0"

Fix
Fixed in line 265 of the file handler.c in the function websUrlHandlerRequest(webs_t wp). Explicitly test that first character of the request is '/' or '\'. Credit: Luigi Auriemma.

Remote shell execution

Issue

A remote attacker can execute a shell on the affected system. In Windows systems, the '\' character can be used as a directory separator. In the cgi module (cgi.c), anti-directory traversal is only implemented for the '/' character (line 70, cgi.c - see below). When you set cgiName to a string with \ separators, you can execute arbitrary commands on the affected machine. Reported 2009-01-26 by Michal Sajdak:

```
HTTP REQUEST:
GET /cgi-bin/..\..\..\..\..\..\..\..\windows\system32\cmd.exe?/k+c:\windows
   \system32\ping.exe+127.0.0.1  HTTP/1.0
HTTP RESPONSE:
HTTP/1.0 200 OK
```

Fix

Whitelist-based URL validation eliminates platform-specific interpretation of URLs. URLs of this type are now rejected. Credit: PeerSec Networks.

Source disclosure

Issue

Windows systems can be tricky when it comes to naming files: trailing dots or spaces are truncated when accessing/creating a file. See: http://msdn.microsoft.com/en-us/library/aa365247.aspx - "Do not end a file or directory name with a trailing space or a period. Although the underlying file system may support such names, the operating system does not. However, it is acceptable to start a name with a period." Reported 2009-01-26 by Michal Sajdak:

When requesting an .asp file, with dot(s) and/or space(s) appended, you get the source of the file. For example, when you make a request to "home.asp...":

1.) The Web server detects that the request is not performed to an .asp file (it does not end with ".asp").

2.) An "open file" request is passed to the operating system, and the operating system strips it to: "home.asp".

3.) home.asp is served as plain text.

Fix

Whitelist-based URL validation eliminates platform-specific interpretation of URLs. URLs of this type are now rejected. Credit: PeerSec Networks

Remote DoS

Issue

Again, http://msdn.microsoft.com/en-us/library/aa365247.aspx is a handy resource. We read there: "Do not use the following reserved device names for the name of a file:

CON, PRN, AUX, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9"

Also, avoid these names followed immediately by an extension; for example, NUL.txt is not recommended." The problem was once addressed in the Web server. There is a protective function called "isBadWindowsPath" which blacklists the following words. Unfortunately, it does not protect against requests in a specific context (accessing a blacklisted *file* - not blacklisted directory):

```
316: if ((badPath(parts[i], T("con"), 3)) || (badPath(parts[i], T("nul"), 3)) ||
    (badPath(parts[i], T("aux"), 3)) ||(badPath(parts[i], T("clock$"), 6)) ||
    (badPath(parts[i], T("config$"), 7)) )
```

Accessing AUX.txt crashes the Web server (http://host/AUX.txt) .Accessing COM1.txt (and maybe other COMs... And LPTs...). (http://host/COM1.txt). Note that COM is not even blacklisted in the isBadWindowsPath function.

Reported 2009-01-26 by Michal Sajdak:

Fix

Whitelist-based URL validation eliminates platform-specific interpretation of URLs. URLs of this type are now rejected, as long as an actual file of the name, for example “AUX.txt”, does not exist in the web root directory. Credit: PeerSec Networks.

Additional verified fixes in 2.5

Several bugs had been reported to exist in the previously released version (2.1.8) that actually had been fixed in that version. PeerSec Networks verified through runtime and/or code inspection that the following reported bugs are fixed as of version 2.1.8.

Buffer Overflow with over 64 elements in the URL path
http://www.securiteam.com/securitynews/5MP0C1580W.html

Fixed in default.c:websValidateUrl(). Credit: Dhanwa T.

GoAhead Web Server DoS (AUX)
http://www.securiteam.com/securitynews/5IP0E2K41I.html

Fixed in version 2.1.8. Unable to reproduce this issue in 2.1.8. Related to the issue, Remote DoS, fixed above.

Read arbitrary files from the server running GoAhead (Directory Traversal)
http://www.securiteam.com/securitynews/5RP0I007PG.html

http://www.securiteam.com/securitynews/5QP010U3FS.html

Fixed in default.c: websValidateURL(). Credit: Matt Moore

Cross Site Scripting via 404 messages
http://www.securiteam.com/securitynews/5RP0I007PG.html

Fixed in default.c:websDefaultHandler(). Credit: BgP.

Copyright Information

Trademarks

GoAhead and GoAhead WebServer are registered trademarks of GoAhead Software. All other brand or product names are the trademarks or registered trademarks of their respective holders.

Copyright

Copy Restrictions

The software described in this document may be used and copied only in accordance with the terms of the accompanying license agreement.

GoAhead Software, Inc.

10900 NE 8th Street Suite 1200 Bellevue, WA 98004 +1 (425) 453-1900 www.goahead.com info@goahead.com