

Question 9.1

By instrumenting the binary code of the target program, AFL can track code coverage during execution. Furthermore, AFL uses the feedback obtained from the instrumented binary to mutate input data. This way inputs that lead to the exploration of new code paths are prioritised, which maximizes the effectiveness of the fuzzing process.

Thus, in order for AFL to efficiently and successfully find bugs and vulnerabilities, the binary needs to be instrumented.

```
[*] Validating target binary...
[-] Looks like the target binary is not instrumented! The fuzzer depends on
compile-time instrumentation to isolate interesting test cases while
mutating the input data. For more information, and for tips on how to
instrument binaries, please see /usr/share/doc/afl/README.

When source code is not available, you may be able to leverage QEMU
mode support. Consult the README for tips on how to enable this.
(It is also possible to use afl-fuzz as a traditional, "dumb" fuzzer.
For that, you can use the -n option - but expect much worse results.)

[-] PROGRAM ABORT : No instrumentation detected
  Location : check_binary(), afl-fuzz.c:6843
```

Question 9.2

AFL still "fails" because it does nothing. It fails to start fuzzing and find crashes.

```
[-] SYSTEM ERROR : Unable to create '.build.afl/generated/board.map'
  Stop location : write_to testcase(), afl-fuzz.c:2469
    OS message : No such file or directory
user@debian:~/buildengine$ _
```

Question 9.3

We remove by commenting out the following:

_platform_init(argc, argv, "BUILD editor by Ken Silverman", "BUILD");

```
GNU nano 2.7.4                               File: build.c                         Modified
                                              File: build.c                         Modified
                                              Modified
case 1: case 2: ExtEditSectorData(searchsector); break;
case 0: case 4: ExtEditWallData(searchwall); break;
case 3: ExtEditSpriteData(searchwall); break;
}
keystatus[0x41] = 0, keystatus[0x42] = 0;
}

if (keystatus[buildkeys[14]] > 0) /* Enter */
{
    overheadeditor();
    keystatus[buildkeys[14]] = 0;
}

int main(int argc,char **argv)
{
    char ch, quitflag;
    long i, j, k;
// _platform_init(argc, argv, "BUILD editor by Ken Silverman", "BUILD");
    if (getenv("BUILD_NOPIENTUM") != NULL)
        setmmxoverlay(0);

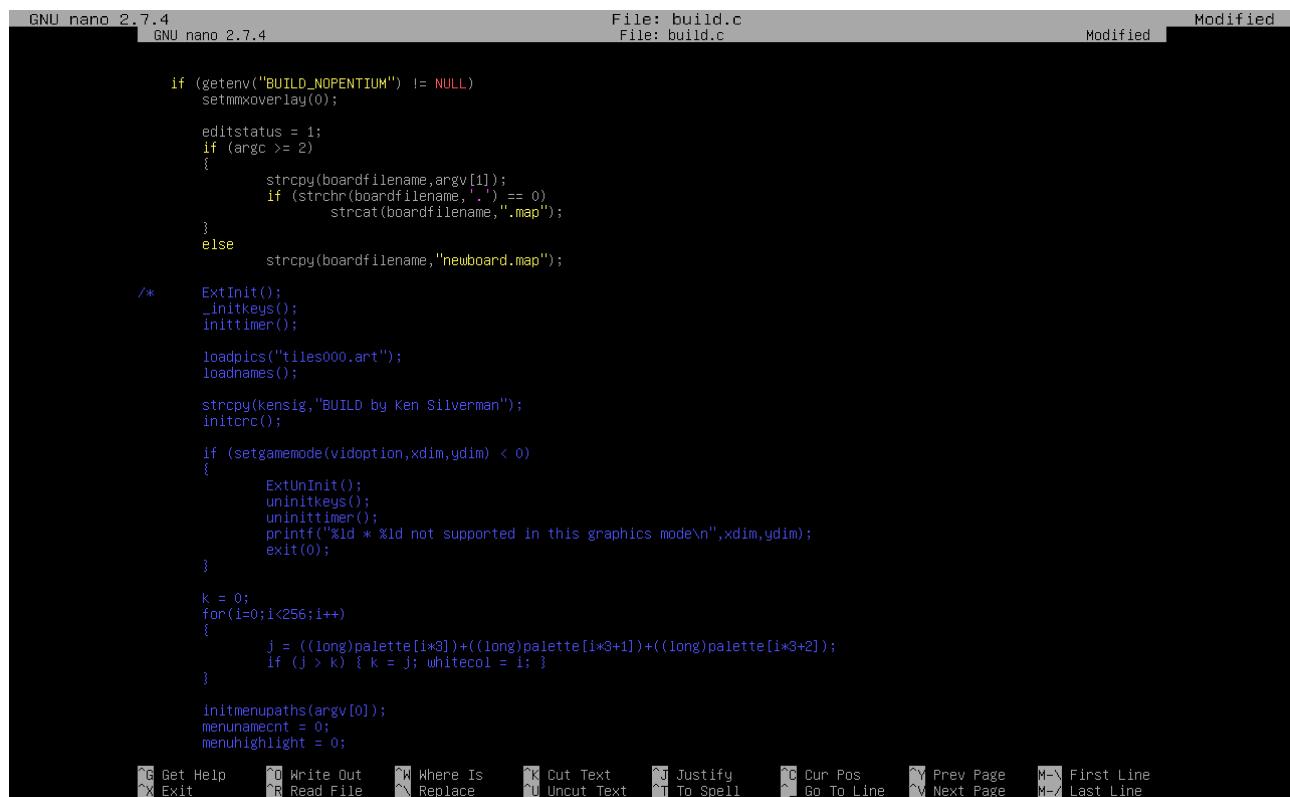
    editstatus = 1;
    if (argc >= 2)
    {
        strcpy(boardfilename,argv[1]);
        if (strchr(boardfilename,'.') == 0)
            strcat(boardfilename,".map");
    }
    else
        strcpy(boardfilename,"newboard.map");

/*     ExtInit();
     _InitKeys();
     InitTimer();

     loadpics("tile000.art");
     loadnames();
```

[G] Get Help [O] Write Out [W] Where Is [K] Cut Text [J] Justify [C] Cur Pos [V] Prev Page [M-\] First Line
 [X] Exit [R] Read File [R] Replace [U] Uncut Text [T] To Spell [G] Go To Line [N] Next Page [M-\] Last Line

As well as, we remove by commenting out everything shown in blue from the following screenshots:



```

GNU nano 2.7.4                               File: build.c                         Modified
GNU nano 2.7.4

if (getenv("BUILD_NOPENTIUM") != NULL)
    setmmxoverlay(0);

editstatus = 1;
if (argc >= 2)
{
    strcpy(boardfilename,argv[1]);
    if (strchr(boardfilename, '.') == 0)
        strcat(boardfilename,".map");
    else
        strcpy(boardfilename,"newboard.map");

/*     ExtInit();
    _initkeys();
    inittimer();

loadpics("tiles000.art");
loadnames();

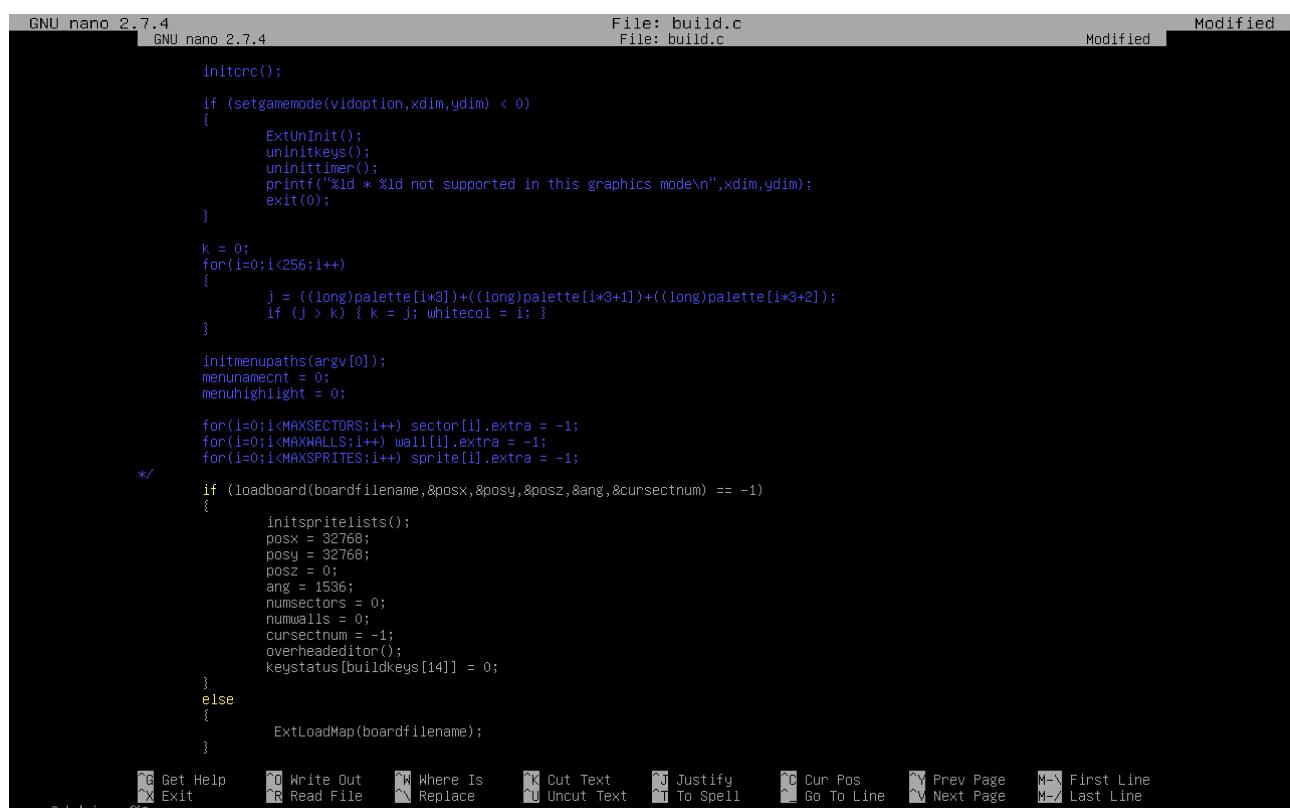
strcpy(kensig,"BUILD by Ken Silverman");
initcrc();

if (setgamemode(vidoption,xdim,ydim) < 0)
{
    ExtUnInit();
    uninitkeys();
    uninittimer();
    printf("%d * %d not supported in this graphics mode\n",xdim,ydim);
    exit(0);
}

k = 0;
for(i=0;i<256;i++)
{
    j = ((long)palette[i*3])+((long)palette[i*3+1])+((long)palette[i*3+2]);
    if (j > k) { k = j; whitecol = i; }
}

initmenupaths(argv[0]);
menunamecnt = 0;
menuhighlight = 0;

^K Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   ^Y Prev Page   M-\ First Line
^X Exit      ^R Read File   ^N Replace    ^U Uncut Text  ^I To Spell   ^L Go To Line  ^V Next Page   M-/ Last Line
user@debian:~$
```

```

GNU nano 2.7.4                               File: build.c                         Modified
GNU nano 2.7.4

initcrc();

if (setgamemode(vidoption,xdim,ydim) < 0)
{
    ExtUnInit();
    uninitkeys();
    uninittimer();
    printf("%d * %d not supported in this graphics mode\n",xdim,ydim);
    exit(0);
}

k = 0;
for(i=0;i<256;i++)
{
    j = ((long)palette[i*3])+((long)palette[i*3+1])+((long)palette[i*3+2]);
    if (j > k) { k = j; whitecol = i; }
}

initmenupaths(argv[0]);
menunamecnt = 0;
menuhighlight = 0;

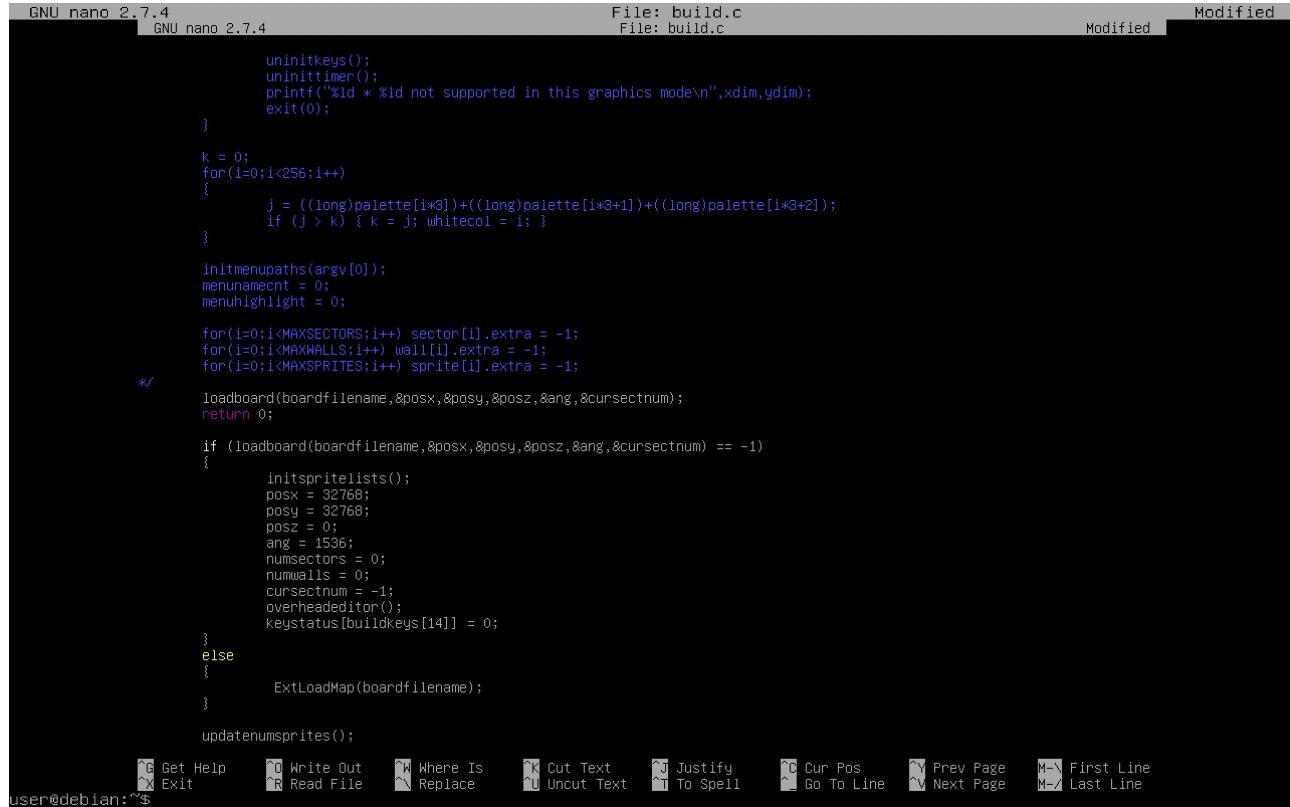
for(i=0;i<MAXSECTORS;i++) sector[i].extra = -1;
for(i=0;i<MAXWALLS;i++) wall[i].extra = -1;
for(i=0;i<MAXSPRITES;i++) sprite[i].extra = -1;
*/
if (loadboard(boardfilename,&posx,&posy,&posz,&ang,&cursectnum) == -1)
{
    initspritelists();
    posx = 32768;
    posy = 32768;
    posz = 0;
    ang = 1536;
    numsectors = 0;
    numwalls = 0;
    cursectnum = -1;
    overheadeditor();
    keystatus[buildkeys[14]] = 0;
}
else
{
    ExtLoadMap(boardfilename);
}

^K Get Help   ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   ^Y Prev Page   M-\ First Line
^X Exit      ^R Read File   ^N Replace    ^U Uncut Text  ^I To Spell   ^L Go To Line  ^V Next Page   M-/ Last Line
user@debian:~$
```

Question 9.4

We update the code to quit the program once the code parsing the map file has finished by putting “return 0;” after the call to

```
loadboard(boardfilename,&posx,&posy,&posz,&ang,&cursectnum);
```



```

GNU nano 2.7.4                               File: build.c
GNU nano 2.7.4                               File: build.c
Modified

        uninitkeys();
        uninittimer();
        printf("%d * %d not supported in this graphics mode\n",xdim,ydim);
        exit(0);
    }

    k = 0;
    for(i=0;i<256;i++)
    {
        j = ((long)palette[i*3]) + ((long)palette[i*3+1]) + ((long)palette[i*3+2]);
        if (j > k) { k = j; whitecol = i; }
    }

    initmenupaths(argv[0]);
    menunamecnt = 0;
    menuhighlight = 0;

    for(i=0;i<MAXSECTORS;i++) sector[i].extra = -1;
    for(i=0;i<MAXWALLS;i++) wall[i].extra = -1;
    for(i=0;i<MAXSPRITES;i++) sprite[i].extra = -1;
/*
    loadboard(boardfilename,&posx,&posy,&posz,&ang,&cursectnum);
    return 0;
}

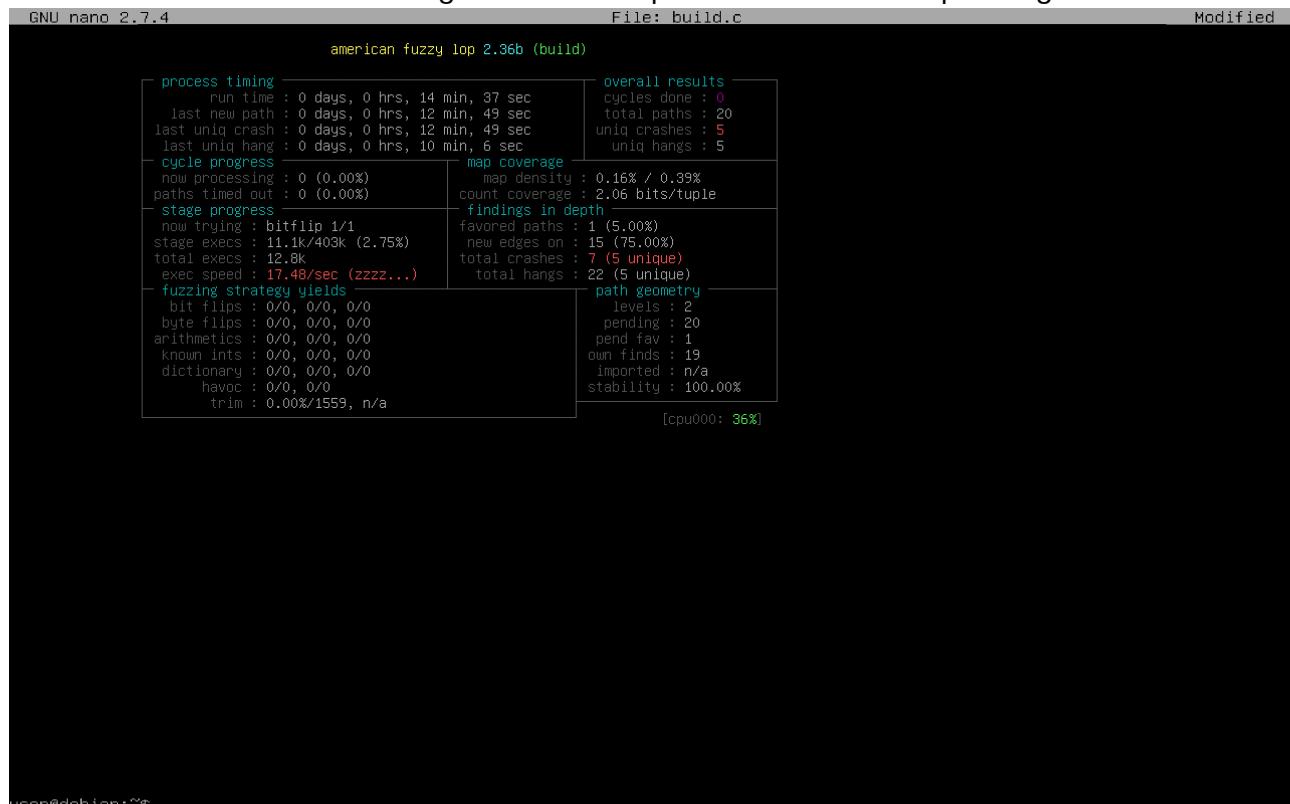
if (loadboard(boardfilename,&posx,&posy,&posz,&ang,&cursectnum) == -1)
{
    initspritelists();
    posx = 32768;
    posy = 32768;
    posz = 0;
    ang = 1536;
    numsectors = 0;
    numwalls = 0;
    cursectnum = -1;
    overheadeditor();
    keystatus[buildkeys[14]] = 0;
}
else
{
    ExtLoadMap(boardfilename);
}

updatenumsprites();

^G Get Help      ^O Write Out      ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos      ^Y Prev Page      M-> First Line
^X Exit          ^R Read File      ^L Replace      ^U Uncut Text     ^I To Spell      ^D Go To Line     ^V Next Page      M-> Last Line
user@debian:~$
```

Question 9.5

We let AFL run for 14 minutes. It generates 5 unique crashes and 5 unique hangs:



process timing		overall results	
run time : 0 days, 0 hrs, 14 min, 37 sec	last new path : 0 days, 0 hrs, 12 min, 49 sec	cycles done : 0	total paths : 20
last uniq crash : 0 days, 0 hrs, 12 min, 49 sec	last uniq hang : 0 days, 0 hrs, 10 min, 6 sec	uniq crashes : 5	uniq hangs : 5
cycle progress		map coverage	
now processing : 0 (0.00%)	paths timed out : 0 (0.00%)	map density : 0.16% / 0.39%	count coverage : 2.06 bits/tuple
stage progress		findings in depth	
now trying : bitflip 1/1	stage execs : 11.1k/403k (2.75%)	favored paths : 1 (5.00%)	new edges on : 15 (75.00%)
total execs : 12.8k	exec speed : 17.48/sec (zzzz...)	total crashes : 7 (5 unique)	total hangs : 22 (5 unique)
fuzzing strategy yields		path geometry	
bit flips : 0/0, 0/0, 0/0	byte flips : 0/0, 0/0, 0/0	levels : 2	pending : 20
arithmetics : 0/0, 0/0, 0/0	known ints : 0/0, 0/0, 0/0	pend fav : 1	own finds : 19
dictionary : 0/0, 0/0, 0/0	havoc : 0/0, 0/0	imported : n/a	stability : 100.00%
trim : 0.00%/1559, n/a			

[cpu000: 36%]

user@debian:~\$

Question 9.6

```

GNU nano 2.7.4                               File: build.c                               Modified
[1]+  Terminated                  ./script.sh
user@debian:~/buildengine$ ls
a.asm      bstub.o    buildper1.c dos_drvr.c   game.c     multi.c    README-dosbins.txt unix_compat.c
a.c       build      buildper1.h doxyfile   game.o     names.h    README-win32bins.txt unix_compat.h
a.gnu.c    build2.txt  build.sh    engine.c   game.pl    nsmoal.map  script.sh    unix_compat.o
a.h       build.afl  build.txt   Engine.dsp  K.asm     nukeland.map sdi_driver.c  utils
a.nasm.asm build.c    cacheid.c engine.h   kdmeng.c  platform.h sdi_driver.o  ves2.h
a.o       buildgl.c cacheid.h engine.o   kensiz.map pragmas.c  setup.dat   win32_compat.h
astboard.map buildgl.h cacheid.o engine_protos.h Makefile  pragmas_gnu.c sreadme.txt
a.visualc.c build.h   CHANGELOG Engine.vcproj Makefile.dos pragmas.h  stuff.dat
boards.map buildin.txt CONTRIB   evial.map  Makefile.u032 pragmas.o  test.map
bstub.c    BUILDLCIC.TXT display.h FILEID.DIZ mmulti.c  pragmas_visualc.c tests
bstub.h    build.o   doscpmat.h game      mmulti.o  README    TODO
user@debian:~/buildengine$ cd
user@debian:~$ ls
automount.sh buildengine CFI [REDACTED]
user@debian:~$ cd buildengine
user@debian:~/buildengine$ cd build.afl/findings/crashes
user@debian:~/buildengine/build.afl/findings/crashes$ ls
id:000000,sig:11,src:000000,op:flip1,pos:20
id:000001,sig:11,src:000000,op:flip1,pos:20 id:000004,sig:11,src:000000,op:flip1,pos:21
id:000002,sig:11,src:000000,op:flip1,pos:20 README.txt
user@debian:~/buildengine/build.afl/findings/crashes$


GNU nano 2.7.4                               File: build.c                               Modified
Segmentation fault
user@debian:~/buildengine$ gdb -q ..../build^C
user@debian:~/buildengine$ cd build.afl/findings/crashes/
user@debian:~/buildengine/build.afl/findings/crashes$ gdb -q ..../..../build
Reading symbols from ..../..../build...done.
(gdb) r id:000000,sig:11,src:000000,op:flip1,pos:20
Starting program: /home/user/buildengine/build id:000000,sig:11,src:000000,op:flip1,pos:20
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[Inferior 1 (process 25389) exited normally]
(gdb) quit
user@debian:~/buildengine/build.afl/findings/crashes$ ..../..../build build.afl/findings/crashes/id:000000,sig:11,src:000000
0,,op:flip1,,pos:20
user@debian:~/buildengine/build.afl/findings/crashes$ ..../..../build id:000000,,sig:11,,src:000000,,op:flip1,,pos:20
user@debian:~/buildengine/build.afl/findings/crashes$ ..../..../build id:000000,,sig:11,,src:000000,,op:flip1,,pos:20 ^C
user@debian:~/buildengine/build.afl/findings/crashes$ ..../..../build
user@debian:~/buildengine/build.afl/findings/crashes$ ..../..../build id:000000,,sig:11,,src:000000,,op:flip1,,pos:20
user@debian:~/buildengine/build.afl/findings/crashes$ cd ..
user@debian:~/buildengine/build.afl/findings$ cd ..
user@debian:~/buildengine/build.afl$ cd ..
user@debian:~/buildengine/build.afl$ ./build
Segmentation fault
user@debian:~/buildengine$ gdb ./build
GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type 'apropos word' to search for commands related to "word"...
Reading symbols from ./build...done.
(gdb) r build.afl/findings/crashes/id:000000,sig:11,src:000000,op:flip1,pos:20
Starting program: /home/user/buildengine/build build.afl/findings/crashes/id:000000,sig:11,src:000000,op:flip1,pos:20
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
inside (sectnum=179, y=<optimized out>, x=<optimized out>) at engine.c:4276
4276           y1 = wal->y-y; y2 = wall(wal->point2).y-y;
(gdb) -
user@debian:~$


Debian GNU/LiThe program being debugged has been started already.
Start it from the beginning? (y or n) y
debian login:Starting program: /home/user/buildengine/build build.afl/findings/crashes/id:000001,sig:11,src:000000,op:flip1,pos:20
Password: [Thread debugging using libthread_db enabled]
Last login: Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Linux debian
Program received signal SIGSEGV, Segmentation fault.
The programs 0x5663964c in inside (sectnum=155, y=<optimized out>, x=<optimized out>) at engine.c:4276
the exact dis 4276           y1 = wal->y-y; y2 = wall(wal->point2).y-y;
individual fi The program being debugged has been started already.
Start it from the beginning? (y or n) y
Debian GNU/LiStarting program: /home/user/buildengine/build build.afl/findings/crashes/id:000002,sig:11,src:000000,op:flip1,pos:20
permitted by [Thread debugging using libthread_db enabled]
user@debian:~ Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1". :11,,src:000000,op:flip1,,pos:20
-bash: /home/Program received signal SIGSEGV, Segmentation fault.
user@debian:~/inside (sectnum=148, y=<optimized out>, x=<optimized out>) at engine.c:4276
user@debian:~:4276           y1 = wal->y-y; y2 = wall(wal->point2).y-y; ,op:flip1,,gdb) r build.afl/findings/crashes/id:000003,sig:11,,src:000000,,op:flip1,,pos:20
user@debian:~: The program being debugged has been started already.
user@debian:~ Start it from the beginning? (y or n) y
user@debian:~/Starting program: /home/user/buildengine/build build.afl/findings/crashes/id:000003,sig:11,,src:000000,,op:flip1,,pos:20
GNU gdb (Deb[Thread debugging using libthread_db enabled]
Copyright (C)Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
License GPLv3+
This is free Program received signal SIGSEGV, Segmentation fault.
There is NO Wlinside (sectnum=148, y=<optimized out>, x=<optimized out>) at engine.c:4276
and "show war4276           y1 = wal->y-y; y2 = wall(wal->point2).y-y;
This GDB was (gdb) r build.afl/findings/crashes/id:000004,sig:11,,src:000000,,op:flip1,,pos:21
Type "show cc" The program being debugged has been started already.
For bug reportStart it from the beginning? (y or n) y
<http://www.gStarting program: /home/user/buildengine/build build.afl/findings/crashes/id:000004,sig:11,,src:000000,,op:flip1,,pos:21
Find the GDB [Thread debugging using libthread_db enabled]
<http://www.g Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
For help, typProgram received signal SIGSEGV, Segmentation fault.
Type "aproposinside (sectnum=2195, y=<optimized out>, x=<optimized out>) at engine.c:4276
Reading symb4276           y1 = wal->y-y; y2 = wall(wal->point2).y-y;
(gdb) r build(gdb) bt
Starting prog#0 inside (sectnum=2195, y=<optimized out>, x=<optimized out>) at engine.c:4276
[Thread debug#1 updatesector (x=0x7648, sectnum=0x567329e4 <sectnum>) at engine.c:7099
Using host li#2 0x56630b1 in loadboard (
filename=0x5661fc8 <boardfilename> "build/afl/findings/crashes/id:000004,sig:11,,src:000000,,op:flip1,,pos:21",
Program recei dapsosx=0x567329cc <posx>, dapsosy=0x567329a4 <posy>, daposz=0x5675e788 <posz>, daang=0x56732a04 <ang>,
inside (sectn dacssectnum=0x567329e4 <sectnum>) at engine.c:3046
4276 #3 0x5655781b in main (argc=<optimized out>, argv=<optimized out>) at build.c:6823
(gdb) r build(gdb) info registers_

```

```

Debian GNU/Liinside (sectnum=148, y=<optimized out>, x=<optimized out>) at engine.c:4276
    4276          y1 = wal->y-y; y2 = wall[wal->point2].y-y;
debian login:(gdb) r build.afl/findings/crashes/idx:000003\,sig\::11\,src\::000000\,op\::flip1\,pos\::20
Password: The program being debugged has been started already.
Last login: Start it from the beginning? (y or n) y
Linux debian Starting program: /home/user/buildengine/build build.afl/findings/crashes/idx:000003\,sig\::11\,src\::000000\,op\::flip1\,pos\::20
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
The programs
the exact dis
individual fiProgram received signal SIGSEGV, Segmentation fault.
inside (sectnum=148, y=<optimized out>, x=<optimized out>) at engine.c:4276
Debian GNU/Liinside (gdb) r build.afl/findings/crashes/idx:000004\,sig\::11\,src\::000000\,op\::flip1\,pos\::21
permitted by: The program being debugged has been started already.
user@debian: Start it from the beginning? (y or n) y
000\,op\::flipStarting program: /home/user/buildengine/build build.afl/findings/crashes/idx:000004\,sig\::11\,src\::000000\,op\::flip1\,pos\::21
-bash: /home/[Thread debugging using libthread_db enabled]
user@debian:Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
user@debian:\
\,op\::flip1\,Program received signal SIGSEGV, Segmentation fault.
user@debian: inside (sectnum=2195, y=<optimized out>, x=<optimized out>) at engine.c:4276
user@debian:4276          y1 = wal->y-y; y2 = wall[wal->point2].y-y;
user@debian:^(gdb) bt
GNU gdb (Debi#0 inside (sectnum=2195, y=<optimized out>, x=<optimized out>) at engine.c:4276
Copyright (C)1 updatesector (x=27648, y=13568, sectnum=0x567329e4 <currentsectnum>) at engine.c:7099
License GPLv2#2 0x566303b1 in loadboard (
This is free filename="boardfilename" "build.afl/findings/crashes/idx:000004,sig:11,src:000000,op:flip1,pos:21",
There is NO daposx=0x567329c <posx>, daposy=0x567329a4 <posy>, daposz=0x5675e788 <posz>, daang=0x56732a04 <ang>,
and 'show war dacurrentsectnum=0x567329e4 <currentsectnum>' at engine.c:3046
This GDB was #3 0x5655781b in main (argc=<optimized out>, argv=<optimized out>) at build.c:6823
Type "show cc"(gdb) info registers
For bug reportax 0x567f01e0 1451164128
<http://www.gdbx 0x3a00 14848
Find the GDB edx 0x56887ce0 1451785440
<http://www.gbx 0xfffffc00 -13568
For help, typebp 0xfffffd550 0xfffffd550
Type "aproposbp Reading symboldi 0x3a00 14848
(gdb) r buildelp 0x3500 13568
Starting progelags 0x56639741 0x56639741 <updatesector+5809>
[Thread debugcs 0x20 [ IF ]
Using host liss 0x23 35
ds 0x2b 43
Program receives 0x2b 43
inside (sectnfs 0x0 0
4276 gs 0x63 99
(gdb) r build(gdb)

```

The crash happens at line 4276 of code engine.c, which is found in the following line:

```

GNU nano 2.7.4                               File: build.c
GNU nano 2.7.4                               File: ../.././engine.c
Modified
if ((i >= startumost[j]) && (i < startdmst[j]))
    drawpixel(j+p,col);
    plc += inc; p += ylookup[i];
}
}

int inside(long x, long y, short sectnum)
{
    walltype *wal;
    long i, x1, y1, x2, y2;
    unsigned long cnt;

    if ((sectnum < 0) || (sectnum >= numsectors)) return(-1);

    cnt = 0;
    wal = &wall[sector[sectnum].wallptr];
    i = sector[sectnum].wallnum;
    do
    {
        y1 = wal->y-y; y2 = wall[wal->point2].y-y;
        if ((y1^y2) < 0)
        {
            x1 = wal->x-x; x2 = wall[wal->point2].x-x;
            if (((x1^x2) >= 0) cnt ^= x1; else cnt ^= (x1*x2-x2*x1)^y2;
        }
        wal++; i--;
    } while (i);
    return(cnt>31);
}

int getangle(long xvect, long yvect)
{
    if ((xvect|yvect) == 0) return(0);
    if (xvect == 0) return(512+((yvect<0)<<10));
    if (yvect == 0) return(((xvect<0)<<10));
    if (xvect == yvect) return(256+((xvect<0)<<10));
    if (xvect == -yvect) return(768+((xvect<0)<<10));
    if (klabs(xvect) > klabs(yvect))
        return(((radarang[640+scale(160,yvect,xvect)]>>6)+((xvect<0)<<10))&2047);
    return(((radarang[640-scale(160,xvect,yvect)]>>6)+512+((yvect<0)<<10))&2047);

^K Get Help   ^O Write Out   ^W Where Is   ^X Cut Text   ^J Justify   ^C Cur Pos   ^Y Prev Page   M-\ First Line
^X Exit      ^R Read File   ^\ Replace   ^U Uncut Text  ^T To Spell   ^G Go To Line  ^V Next Page   M-/ Last Line
user@debian:~$
```

Let's examine the function *inside*. We see it uses variables *sectnum*, *wallnum*, *wall*, and *sector*.

Let's examine those variables in *build.h*:

```
Debian GNU/Li  GNU nano 2.7.4                               File: build.h

debian login:4*
Password: * "Build Engine & Tools" Copyright (c) 1993-1997 Ken Silverman
Last login: T * Ken Silverman's official web site: "http://www.advsys.net/ken"
Linux debian * See the included license file "BUILDLOC.TXT" for license info.
Linux debian * This file has been modified from Ken Silverman's original release
The programs */
the exact dis
individual fi #ifndef _INCLUDE_BUILD_H_
#define _INCLUDE_BUILD_H_

Debian GNU/Li #define MAXSECTORS 1024
permitted by #define MAXWALLS 8192
user@debian: #define MAXSPRITES 4096
000\,op:flip :11\,src\,000
-bash: /home/ #define MAXTILES 9216
user@debian: #define MAXSTATUS 1024
user@debian: #define MAXPLAYERS 16
\,op:flip1\, #define MAXDIM 1600
user@debian: #define MAXDIM 1200
user@debian: #define MAXPALOOKUPS 256
user@debian: #define MAXSKYTILES 256
GNU gdb (Debi) #define MAXSPRITESONSCREEN 1024
Copyright (C)
License GPLv3#define CLIPMASK0 (((1L)<<16)+1L)
This is free #define CLIPMASK1 (((256L)<<16)+64L)
There is NO W
and 'show war
This GDB was      /*
Type "show co   * Make all variables in BUILD.H defined in the ENGINE,
For bug repor   * and externed in GAME
<http://www.g   /*
Find the GDB #ifdef ENGINE
<http://www.g   #define EXTERN
For help, typ #else
Type "apropos  #define EXTERN extern
Reading symb #endif
(gdb) r build
Starting prog
[Thread debug#ifdef PLATFORM_DOS
Using host li #pragma pack(push,1);
 #else
Program recei#pragma pack(1)
inside (sectn [ Read 314 lines ]
4276  ^G Get Help    ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify
(gdb) r build^X Exit    ^R Read File   ^\ Replace   ^U Uncut Text  ^I To Spell
                                         ^C Cur Pos   ^Y Prev Page  M-> First Line
                                         ^V Go To Line ^N Next Page  M-> Last Line
```

```
Debian GNU/Li  GNU nano 2.7.4                               File: build.h

debian login:4*
Password: /* 44 bytes */
Last login: T #typedef struct
Linux debian {
    long x, y, z;
    short cstat, picnum;
    signed char shade;
    unsigned char pal, clipdist, filler;
    unsigned char xrepeat, yrepeat;
    signed char xoffset, yoffset;
    short sectnum, statnum;
    short ang, owner, xvel, yvel, zvel;
    short lotag, hitag, extra;
000\,op:flip} sprite-type;
-bash: /home/ #ifdef PLATFORM_DOS
user@debian: #pragma pack(pop);
\,op:flip1\, #else
user@debian: #pragma pack()
user@debian: #endif
user@debian: #EXTERN sector-type sector[MAXSECTORS];
Copyright (C)EXTERN wall-type wall[MAXWALLS];
License GPLv3EXTERN sprite-type sprite[MAXSPRITES];
This is free
There is NO WEXTERN long spritesortcnt;
and 'show warEXTERN sprite-type tsprite[MAXSPRITESONSCREEN];
This GDB was
Type "show coEXTERN char vidoption;
For bug reporEXTERN long xdim, ydim, ylookup[MAXYDIM+1], numpages;
<http://www.gEXTERN long yxaspect, viewingrange;
Find the GDB #EXTERN long validmodcnt;
<http://www.gEXTERN short validmode[256];
For help, typEXTERN long validmodexdim[256], validmodeydim[256];
Type "apropos
Reading symb #EXTERN short numsectors, numwalls;
(gdb) r build#EXTERN volatile long totalclock;
Starting prog#EXTERN long numframes, randomseed;
[Thread debug#EXTERN short sintable[2048];
Using host li#EXTERN unsigned char palette[768];
    #EXTERN short numpalookups;
Program recei#EXTERN char *palookup[MAXPALOOKUPS];
inside (sectn [ Read 314 lines ]
4276  ^G Get Help    ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify
(gdb) r build^X Exit    ^R Read File   ^\ Replace   ^U Uncut Text  ^I To Spell
                                         ^C Cur Pos   ^Y Prev Page  M-> First Line
                                         ^V Go To Line ^N Next Page  M-> Last Line
```

```

Debian GNU/Li  GNU nano 2.7.4          File: build.h

debian login: *      00 = normal floors
Password: *      01 = masked floors
Last login: T      10 = translucent masked floors
Linux debian *      11 = reverse translucent masked floors
               *      bits 9-15: reserved
The programs */
the exact dis
individual fi /* 40 bytes */
typedef struct
Debian GNU/Li{
    short wallptr, wallnum;
permitted by
user@debian:~ long ceilingz, floorz;
short ceilingstat, floorstat;
short ceilingpicnum, ceilingheinum;
-bash: /home/
signed char ceilingshade;
user@debian:~ unsigned char ceilingpal, ceilingxpanning, ceilingypanning;
user@debian:~ short floorpicnum, floorheinum;
\,op:flip1\, signed char floorshade;
user@debian:~ unsigned char floorpal, floorxpanning, floorypanning;
user@debian:~ unsigned char visibility, filler;
user@debian:~ short lotag, hitag, extra;
GNU gdb (Debian) sectorstype;
Copyright (C)
License GPLv3/*
This is free */
cstat:
There is NO W * bit 0: 1 = Blocking wall (use with clipmove, getzrange)
and 'show war * bit 1: 1 = bottoms of invisible walls swapped, 0 = not
This GDB was * bit 2: 1 = align picture on bottom (for doors), 0 = top
Type "show co * bit 3: 1 = x-flipped, 0 = normal
For bug repor * bit 4: 1 = masking wall, 0 = not
For http://www.g * bit 5: 1 = 1-way wall, 0 = not
Find the GDB * bit 6: 1 = Blocking wall (use with hitscan / cliptype 1)
 * bit 7: 1 = Translucence, 0 = not
For help, typ * bit 8: 1 = y-flipped, 0 = normal
Type "apropos * bit 9: 1 = Translucence reversing, 0 = normal
Reading symb */
(gdb) r build
Starting prog /* 32 bytes */
[Thread debugtypedef struct
Using host li{
    long x, y;
Program recei
inside (sectn
4276  ^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  M-> First Line
(gdb) r build^X Exit  ^R Read File  ^N Replace  ^U Uncut Text  ^T To Spell  ^G Go To Line  ^V Next Page  M-> Last Line

```

```

Debian GNU/Li  GNU nano 2.7.4          File: build.h

debian login: * bit 1: 1 = bottoms of invisible walls swapped, 0 = not      "2"
Password: * bit 2: 1 = align picture on bottom (for doors), 0 = top      "0"
Last login: T      bit 3: 1 = x-flipped, 0 = normal      "F"
Linux debian * bit 4: 1 = masking wall, 0 = not      "M"
               * bit 5: 1 = 1-way wall, 0 = not      "1"
The programs * bit 6: 1 = Blocking wall (use with hitscan / cliptype 1)      "H"
the exact dis * bit 7: 1 = Translucence, 0 = not      "T"
individual fi * bit 8: 1 = y-flipped, 0 = normal      "F"
               * bit 9: 1 = Translucence reversing, 0 = normal      "T"
bits 10-15: reserved
Debian GNU/Li */
permitted by
user@debian:~ /* 32 bytes */
000\,op\,fliptypedef struct
-bash: /home/
user@debian:~ long x, y;
user@debian:~ short point2, nextwall, nextsector, cstat;
\,op\,flip1\, short picnum, overpicnum;
user@debian:~ signed char shade;
user@debian:~ char pal, xrepeat, yrepeat, xpanning, ypanning;
user@debian:~ short lotag, hitag, extra;
GNU gdb (Debian) walltype;
Copyright (C)
License GPLv3
This is free */
There is NO W * cstat:
and "show war * bit 0: 1 = Blocking sprite (use with clipmove, getzrange)      "B"
This GDB was * bit 1: 1 = translucence, 0 = normal      "T"
Type "show co * bit 2: 1 = x-flipped, 0 = normal      "F"
For bug repor * bit 3: 1 = y-flipped, 0 = normal      "F"
For http://www.g * bits 5-4: 00 = FACE sprite (default)      "R"
Find the GDB *      01 = WALL sprite (like masked walls)
 *      10 = FLOOR sprite (parallel to ceilings&floors)
For help, typ * bit 6: 1 = 1-sided sprite, 0 = normal      "1"
Type "apropos * bit 7: 1 = Real centered centering, 0 = foot center      "C"
Reading symb */
(gdb) r build * bit 8: 1 = Blocking sprite (use with hitscan / cliptype 1)      "H"
               * bit 9: 1 = Translucence reversing, 0 = normal      "T"
bits 10-14: reserved
Starting prog * bit 15: 1 = Invisible sprite, 0 = not invisible      "I"
[Thread debug */
Using host li /* 44 bytes */
Program recei
inside (sectn
4276  ^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  ^Y Prev Page  M-> First Line
(gdb) r build^X Exit  ^R Read File  ^N Replace  ^U Uncut Text  ^T To Spell  ^G Go To Line  ^V Next Page  M-> Last Line

```

We can see that we have global buffers *sector* and *wall*. The *sector* variable can hold a maximum of 1024 *sector* structures, while a *sector* structure represents 40 bytes. The *wall* variable can hold a maximum of 8192 *wall* structures, while a *wall* structure represents 32 bytes. If an attacker can control the number of bytes to copy to the global buffers, there would be a buffer overflow vulnerability.

Furthermore, we use (*gdb*) *info proc mappings* to check the permissions of the stack. However, the permissions of the stack are not mentioned explicitly, as seen in the screenshot below. That's why we attempt to write to the stack:

```

Debian GNU/Li      0xf795e000 0xf7a26000 0xc8000      0x0 /usr/lib/i386-linux-gnu/libcaca.so.0.99.19
          0xf7a26000 0xf7a27000 0x1000      0xc7000 /usr/lib/i386-linux-gnu/libcaca.so.0.99.19
debian login:    0xf7a27000 0xf7a28000 0x1000      0xc8000 /usr/lib/i386-linux-gnu/libcaca.so.0.99.19
Password:        0xf7a28000 0xf7a3b000 0x13000      0x0 /usr/lib/i386-linux-gnu/libXext.so.6.4.0
Last login: T    0xf7a3b000 0xf7a3c000 0x1000      0x12000 /usr/lib/i386-linux-gnu/libXext.so.6.4.0
Linux debian     0xf7a3c000 0xf7a3d000 0x1000      0x13000 /usr/lib/i386-linux-gnu/libXext.so.6.4.0
---Type <return> to continue, or q <return> to quit---
The programs      0xf7a3d000 0xf7b86000 0x149000      0x0 /usr/lib/i386-linux-gnu/libX11.so.6.3.0
the exact dis    0xf7b86000 0xf7b87000 0x1000      0x149000 /usr/lib/i386-linux-gnu/libX11.so.6.3.0
individual fi    0xf7b87000 0xf7b88000 0x1000      0x149000 /usr/lib/i386-linux-gnu/libX11.so.6.3.0
          0xf7b88000 0xf7b89000 0x3000      0x14a000 /usr/lib/i386-linux-gnu/libX11.so.6.3.0
Debian GNU/Li     0xf7b89000 0xf7be3000 0x58000      0x0 /usr/lib/i386-linux-gnu/libpulse.so.0.20.1
permitted by     0xf7be3000 0xf7be4000 0x1000      0x57000 /usr/lib/i386-linux-gnu/libpulse.so.0.20.1
user@debian:~    0xf7be4000 0xf7be5000 0x1000      0x58000 /usr/lib/i386-linux-gnu/libpulse.so.0.20.1 :11\,src\::00000
000\,op:flip    0xf7be5000 0xf7be9000 0x4000      0x0 /usr/lib/i386-linux-gnu/libpulse-simple.so.0.1.0
-bash: /home/   0xf7be9000 0xf7bea000 0x1000      0x3000 /usr/lib/i386-linux-gnu/libpulse-simple.so.0.1.0
user@debian:~    0xf7bea000 0xf7beb000 0x1000      0x4000 /usr/lib/i386-linux-gnu/libpulse-simple.so.0.1.0
user@debian:~    0xf7beb000 0xf7bee000 0x3000      0x0 /lib/i386-linux-gnu/libdl-2.24.so
user@debian:~    0xf7bee000 0xf7bef000 0x1000      0x2000 /lib/i386-linux-gnu/libdl-2.24.so
\,op:flip1\,    0xf7bef000 0xf7bf0000 0x1000      0x3000 /lib/i386-linux-gnu/libdl-2.24.so
user@debian:~    0xf7bf0000 0xf7c43000 0x53000      0x0 /lib/i386-linux-gnu/libm-2.24.so
user@debian:~    0xf7c43000 0xf7c44000 0x1000      0x52000 /lib/i386-linux-gnu/libm-2.24.so
user@debian:~    0xf7c44000 0xf7c45000 0x1000      0x53000 /lib/i386-linux-gnu/libm-2.24.so
GNU gdb (Debi   0xf7c45000 0xf7d63000 0x11e000      0x0 /usr/lib/i386-linux-gnu/libasound.so.2.0.0
Copyright (C)  0xf7d63000 0xf7d68000 0x5000      0x11d000 /usr/lib/i386-linux-gnu/libasound.so.2.0.0
License GPLv3  0xf7d68000 0xf7d69000 0x1000      0x122000 /usr/lib/i386-linux-gnu/libasound.so.2.0.0
This is free   0xf7d69000 0xf7f1a000 0x1b1000      0x0 /lib/i386-linux-gnu/libc-2.24.so
There is NO W  0xf7f1a000 0xf7f1b000 0x1000      0x1b1000 /lib/i386-linux-gnu/libc-2.24.so
and 'show war  0xf7f1b000 0xf7f1d000 0x2000      0x1b1000 /lib/i386-linux-gnu/libc-2.24.so
This GDB was  0xf7f1d000 0xf7f1e000 0x1000      0x1b3000 /lib/i386-linux-gnu/libc-2.24.so
Type "show co  0xf7f1e000 0xf7f21000 0x3000      0x0
For bug repor  0xf7f21000 0xf7f97000 0x76000      0x0 /usr/lib/i386-linux-gnu/libSDL-1.2.so.0.11.4
<http://www.g  0xf7f97000 0xf7f98000 0x1000      0x75000 /usr/lib/i386-linux-gnu/libSDL-1.2.so.0.11.4
Find the GDB   0xf7f98000 0xf7f99000 0x1000      0x76000 /usr/lib/i386-linux-gnu/libSDL-1.2.so.0.11.4
<http://www.g  0xf7f99000 0xf7fc3000 0x2a000      0x0
For help, typ  0xf7fd3000 0xf7fd5000 0x2000      0x0
Type "apropos  0xf7fd5000 0xf7fd7000 0x2000      0x0 [vvar]
Reading symbo  0xf7fd7000 0xf7fd9000 0x2000      0x0 [vdso]
(gdb) r build  0xf7fd9000 0xf7ffc000 0x23000      0x0 /lib/i386-linux-gnu/ld-2.24.so
Starting prog  0x7ffcc000 0xf7fd000 0x1000      0x22000 /lib/i386-linux-gnu/ld-2.24.so
[Thread debug  0x7fffd000 0xf7fe000 0x1000      0x23000 /lib/i386-linux-gnu/ld-2.24.so
Using host li  0x7fffd000 0xfffffe000 0x21000      0x0 [stack] p1\,pos\::20
(gdb) xx $sp
0xfffffd550: 0x566af000
Program recei(gdb) set *(char **)$sp = 'A'
inside (sectr(gdb) xx $sp
4276 0xfffffd550: 0x566af041
(gdb) r build(gdb)

```

We can see that the stack is writable. Therefore, an attacker can overwrite the stack to redirect the control flow to their own code. Thus, the crash is exploitable. The attacker can generate a map to trigger the overflow to overwrite the return address on the stack. This way, the attacker can execute arbitrary code.