

bomb: file format elf64-x86-64

Disassembly of section .init:

0000000000001000 <_init>:

```
1000:    f3 0f 1e fa    endbr64
1004:    48 83 ec 08    sub    $0x8,%rsp
1008:    48 8b 05 d9 3f 00 00    mov    0x3fd9(%rip),%rax    # 4fe8
<__gmon_start__>
100f:    48 85 c0        test   %rax,%rax
1012:    74 02          je     1016 <_init+0x16>
1014:    ff d0          call  *%rax
1016:    48 83 c4 08    add    $0x8,%rsp
101a:    c3            ret
```

Disassembly of section .plt:

0000000000001020 <getenv@plt-0x10>:

```
1020:    ff 35 e2 3f 00 00    push   0x3fe2(%rip)    # 5008
<_GLOBAL_OFFSET_TABLE_+0x8>
1026:    ff 25 e4 3f 00 00    jmp    *0x3fe4(%rip)    # 5010
<_GLOBAL_OFFSET_TABLE_+0x10>
102c:    0f 1f 40 00        nopl   0x0(%rax)
```

0000000000001030 <getenv@plt>:

```
1030:    ff 25 e2 3f 00 00    jmp    *0x3fe2(%rip)    # 5018 <getenv@GLIBC_2.2.5>
1036:    68 00 00 00 00      push   $0x0
103b:    e9 e0 ff ff        jmp    1020 <_init+0x20>
```

0000000000001040 <strcasecmp@plt>:

```
1040:    ff 25 da 3f 00 00    jmp    *0x3fda(%rip)    # 5020
<strcasecmp@GLIBC_2.2.5>
1046:    68 01 00 00 00      push   $0x1
104b:    e9 d0 ff ff        jmp    1020 <_init+0x20>
```

0000000000001050 <__errno_location@plt>:

```
1050:    ff 25 d2 3f 00 00    jmp    *0x3fd2(%rip)    # 5028
<__errno_location@GLIBC_2.2.5>
1056:    68 02 00 00 00      push   $0x2
105b:    e9 c0 ff ff        jmp    1020 <_init+0x20>
```

0000000000001060 <strcpy@plt>:

1060:	ff 25 ca 3f 00 00	jmp	*0x3fca(%rip)	# 5030 <strcpy@GLIBC_2.2.5>
1066:	68 03 00 00 00	push	\$0x3	
106b:	e9 b0 ff ff ff	jmp	1020 <_init+0x20>	

0000000000001070 <puts@plt>:

1070:	ff 25 c2 3f 00 00	jmp	*0x3fc2(%rip)	# 5038 <puts@GLIBC_2.2.5>
1076:	68 04 00 00 00	push	\$0x4	
107b:	e9 a0 ff ff ff	jmp	1020 <_init+0x20>	

0000000000001080 <write@plt>:

1080:	ff 25 ba 3f 00 00	jmp	*0x3fba(%rip)	# 5040 <write@GLIBC_2.2.5>
1086:	68 05 00 00 00	push	\$0x5	
108b:	e9 90 ff ff ff	jmp	1020 <_init+0x20>	

0000000000001090 <strlen@plt>:

1090:	ff 25 b2 3f 00 00	jmp	*0x3fb2(%rip)	# 5048 <strlen@GLIBC_2.2.5>
1096:	68 06 00 00 00	push	\$0x6	
109b:	e9 80 ff ff ff	jmp	1020 <_init+0x20>	

00000000000010a0 <__stack_chk_fail@plt>:

10a0:	ff 25 aa 3f 00 00	jmp	*0x3faa(%rip)	# 5050
-------	-------------------	-----	---------------	--------

<__stack_chk_fail@GLIBC_2.4>

10a6:	68 07 00 00 00	push	\$0x7	
10ab:	e9 70 ff ff ff	jmp	1020 <_init+0x20>	

00000000000010b0 <printf@plt>:

10b0:	ff 25 a2 3f 00 00	jmp	*0x3fa2(%rip)	# 5058 <printf@GLIBC_2.2.5>
10b6:	68 08 00 00 00	push	\$0x8	
10bb:	e9 60 ff ff ff	jmp	1020 <_init+0x20>	

00000000000010c0 <alarm@plt>:

10c0:	ff 25 9a 3f 00 00	jmp	*0x3f9a(%rip)	# 5060 <alarm@GLIBC_2.2.5>
10c6:	68 09 00 00 00	push	\$0x9	
10cb:	e9 50 ff ff ff	jmp	1020 <_init+0x20>	

00000000000010d0 <close@plt>:

10d0:	ff 25 92 3f 00 00	jmp	*0x3f92(%rip)	# 5068 <close@GLIBC_2.2.5>
10d6:	68 0a 00 00 00	push	\$0xa	
10db:	e9 40 ff ff ff	jmp	1020 <_init+0x20>	

00000000000010e0 <read@plt>:

```

10e0:    ff 25 8a 3f 00 00      jmp     *0x3f8a(%rip)      # 5070 <read@GLIBC_2.2.5>
10e6:    68 0b 00 00 00        push    $0xb
10eb:    e9 30 ff ff           jmp     1020 <_init+0x20>

```

00000000000010f0 <fgets@plt>:

```

10f0:    ff 25 82 3f 00 00      jmp     *0x3f82(%rip)      # 5078 <fgets@GLIBC_2.2.5>
10f6:    68 0c 00 00 00        push    $0xc
10fb:    e9 20 ff ff           jmp     1020 <_init+0x20>

```

0000000000001100 <strcmp@plt>:

```

1100:    ff 25 7a 3f 00 00      jmp     *0x3f7a(%rip)      # 5080 <strcmp@GLIBC_2.2.5>
1106:    68 0d 00 00 00        push    $0xd
110b:    e9 10 ff ff           jmp     1020 <_init+0x20>

```

0000000000001110 <signal@plt>:

```

1110:    ff 25 72 3f 00 00      jmp     *0x3f72(%rip)      # 5088 <signal@GLIBC_2.2.5>
1116:    68 0e 00 00 00        push    $0xe
111b:    e9 00 ff ff           jmp     1020 <_init+0x20>

```

0000000000001120 <gethostbyname@plt>:

```

1120:    ff 25 6a 3f 00 00      jmp     *0x3f6a(%rip)      # 5090
<gethostbyname@GLIBC_2.2.5>
1126:    68 0f 00 00 00        push    $0xf
112b:    e9 f0 fe ff ff        jmp     1020 <_init+0x20>

```

0000000000001130 <fprintf@plt>:

```

1130:    ff 25 62 3f 00 00      jmp     *0x3f62(%rip)      # 5098 <fprintf@GLIBC_2.2.5>
1136:    68 10 00 00 00        push    $0x10
113b:    e9 e0 fe ff ff        jmp     1020 <_init+0x20>

```

0000000000001140 <fflush@plt>:

```

1140:    ff 25 5a 3f 00 00      jmp     *0x3f5a(%rip)      # 50a0 <fflush@GLIBC_2.2.5>
1146:    68 11 00 00 00        push    $0x11
114b:    e9 d0 fe ff ff        jmp     1020 <_init+0x20>
_init

```

0000000000001150 <__isoc99_sscanf@plt>:

```

1150:    ff 25 52 3f 00 00      jmp     *0x3f52(%rip)      # 50a8
<__isoc99_sscanf@GLIBC_2.7>

```

1156: 68 12 00 00 00 push \$0x12
115b: e9 c0 fe ff ff jmp 1020 <_init+0x20>

0000000000001160 <memmove@plt>:

1160: ff 25 4a 3f 00 00 jmp *0x3f4a(%rip) # 50b0

<memmove@GLIBC_2.2.5>

1166: 68 13 00 00 00 push \$0x13
116b: e9 b0 fe ff ff jmp 1020 <_init+0x20>

0000000000001170 <fopen@plt>:

1170: ff 25 42 3f 00 00 jmp *0x3f42(%rip) # 50b8

<fopen@GLIBC_2.2.5>

1176: 68 14 00 00 00 push \$0x14
117b: e9 a0 fe ff ff jmp 1020 <_init+0x20>

0000000000001180 <sprintf@plt>:

1180: ff 25 3a 3f 00 00 jmp *0x3f3a(%rip) # 50c0

<sprintf@GLIBC_2.2.5>

1186: 68 15 00 00 00 push \$0x15
118b: e9 90 fe ff ff jmp 1020 <_init+0x20>

0000000000001190 <exit@plt>:

1190: ff 25 32 3f 00 00 jmp *0x3f32(%rip) # 50c8

<exit@GLIBC_2.2.5>

1196: 68 16 00 00 00 push \$0x16
119b: e9 80 fe ff ff jmp 1020 <_init+0x20>

00000000000011a0 <connect@plt>:

11a0: ff 25 2a 3f 00 00 jmp *0x3f2a(%rip) # 50d0

<connect@GLIBC_2.2.5>

11a6: 68 17 00 00 00 push \$0x17
11ab: e9 70 fe ff ff jmp 1020 <_init+0x20>

00000000000011b0 <sleep@plt>:

11b0: ff 25 22 3f 00 00 jmp *0x3f22(%rip) # 50d8

<sleep@GLIBC_2.2.5>

11b6: 68 18 00 00 00 push \$0x18
11bb: e9 60 fe ff ff jmp 1020 <_init+0x20>

00000000000011c0 <__ctype_b_loc@plt>:

```

11c0:    ff 25 1a 3f 00 00    jmp    *0x3f1a(%rip)    # 50e0
<__ctype_b_loc@GLIBC_2.3>
11c6:    68 19 00 00 00      push   $0x19
11cb:    e9 50 fe ff ff      jmp    1020 <_init+0x20>

```

00000000000011d0 <socket@plt>:

```

11d0:    ff 25 12 3f 00 00    jmp    *0x3f12(%rip)    # 50e8
<socket@GLIBC_2.2.5>
11d6:    68 1a 00 00 00      push   $0x1a
11db:    e9 40 fe ff ff      jmp    1020 <_init+0x20>

```

Disassembly of section .text:

00000000000011e0 <_start>:

```

11e0:    f3 0f 1e fa          endbr64
11e4:    31 ed               xor    %ebp,%ebp
11e6:    49 89 d1             mov    %rdx,%r9
11e9:    5e                  pop    %rsi
11ea:    48 89 e2             mov    %rsp,%rdx
11ed:    48 83 e4 f0          and    $0xffffffffffff0,%rsp
11f1:    50                  push   %rax
11f2:    54                  push   %rsp
11f3:    4c 8d 05 76 1a 00 00 lea     0x1a76(%rip),%r8    # 2c70
<__libc_csu_fini>
11fa:    48 8d 0d ff 19 00 00 lea     0x19ff(%rip),%rcx    # 2c00
<__libc_csu_init>
1201:    48 8d 3d d1 00 00 00 lea     0xd1(%rip),%rdi     # 12d9 <main>
1208:    ff 15 d2 3d 00 00    call   *0x3dd2(%rip)      # 4fe0
<__libc_start_main@GLIBC_2.2.5>
120e:    f4                  hlt
120f:    90                  nop

```

0000000000001210 <deregister_tm_clones>:

```

1210:    48 8d 3d 29 44 00 00 lea     0x4429(%rip),%rdi    # 5640
<stdout@GLIBC_2.2.5>
1217:    48 8d 05 22 44 00 00 lea     0x4422(%rip),%rax    # 5640
<stdout@GLIBC_2.2.5>
121e:    48 39 f8             cmp    %rdi,%rax
1221:    74 15               je     1238 <deregister_tm_clones+0x28>
1223:    48 8b 05 ae 3d 00 00 mov     0x3dae(%rip),%rax    # 4fd8

```

<_ITM_deregisterTMCloneTable>

```
122a: 48 85 c0      test    %rax,%rax
122d: 74 09         je     1238 <deregister_tm_clones+0x28>
122f: ff e0        jmp     *%rax
1231: 0f 1f 80 00 00 00 00  nopl   0x0(%rax)
1238: c3          ret
1239: 0f 1f 80 00 00 00 00  nopl   0x0(%rax)
```

00000000000001240 <register_tm_clones>:

```
1240: 48 8d 3d f9 43 00 00  lea     0x43f9(%rip),%rdi      # 5640
```

<stdout@GLIBC_2.2.5>

```
1247: 48 8d 35 f2 43 00 00  lea     0x43f2(%rip),%rsi      # 5640
```

<stdout@GLIBC_2.2.5>

```
124e: 48 29 fe      sub     %rdi,%rsi
1251: 48 89 f0      mov     %rsi,%rax
1254: 48 c1 ee 3f   shr     $0x3f,%rsi
1258: 48 c1 f8 03   sar     $0x3,%rax
125c: 48 01 c6      add     %rax,%rsi
125f: 48 d1 fe      sar     %rsi
1262: 74 14         je     1278 <register_tm_clones+0x38>
1264: 48 8b 05 85 3d 00 00  mov     0x3d85(%rip),%rax      # 4ff0
```

<_ITM_registerTMCloneTable>

```
126b: 48 85 c0      test    %rax,%rax
126e: 74 08         je     1278 <register_tm_clones+0x38>
1270: ff e0        jmp     *%rax
1272: 66 0f 1f 44 00 00  nopw   0x0(%rax,%rax,1)
1278: c3          ret
1279: 0f 1f 80 00 00 00 00  nopl   0x0(%rax)
```

00000000000001280 <__do_global_dtors_aux>:

```
1280: f3 0f 1e fa   endbr64
1284: 80 3d dd 43 00 00 00  cmpb   $0x0,0x43dd(%rip)      # 5668
```

<completed.0>

```
128b: 75 33         jne    12c0 <__do_global_dtors_aux+0x40>
128d: 55          push   %rbp
128e: 48 83 3d 62 3d 00 00  cmpq   $0x0,0x3d62(%rip)      # 4ff8
```

<__cxa_finalize@GLIBC_2.2.5>

```
1295: 00          mov     %rsp,%rbp
1296: 48 89 e5      mov     %rsp,%rbp
1299: 74 0d         je     12a8 <__do_global_dtors_aux+0x28>
```

```

129b:    48 8b 3d 66 3e 00 00    mov     0x3e66(%rip),%rdi        # 5108
<__dso_handle>
12a2:    ff 15 50 3d 00 00      call    *0x3d50(%rip)            # 4ff8
<__cxa_finalize@GLIBC_2.2.5>
12a8:    e8 63 ff ff ff         call    1210 <deregister_tm_clones>
12ad:    c6 05 b4 43 00 00 01    movb    $0x1,0x43b4(%rip)        # 5668
<completed.0>
12b4:    5d                      pop     %rbp
12b5:    c3                      ret
12b6:    66 2e 0f 1f 84 00 00    cs nopw 0x0(%rax,%rax,1)
12bd:    00 00 00
12c0:    c3                      ret
12c1:    66 66 2e 0f 1f 84 00    data16 cs nopw 0x0(%rax,%rax,1)
12c8:    00 00 00 00
12cc:    0f 1f 40 00            nopl    0x0(%rax)

```

00000000000012d0 <frame_dummy>:

```

12d0:    f3 0f 1e fa            endbr64
12d4:    e9 67 ff ff ff         jmp     1240 <register_tm_clones>

```

00000000000012d9 <main>:

```

12d9:    53                      push    %rbx
12da:    83 ff 01                cmp     $0x1,%edi
12dd:    0f 84 f8 00 00 00      je      13db <main+0x102>
12e3:    48 89 f3                mov     %rsi,%rbx
12e6:    83 ff 02                cmp     $0x2,%edi
12e9:    0f 85 1c 01 00 00      jne     140b <main+0x132>
12ef:    48 8b 7e 08             mov     0x8(%rsi),%rdi
12f3:    48 8d 35 0a 1d 00 00    lea     0x1d0a(%rip),%rsi        # 3004
<_IO_stdin_used+0x4>
12fa:    e8 71 fe ff ff         call    1170 <fopen@plt>
12ff:    48 89 05 6a 43 00 00    mov     %rax,0x436a(%rip)        # 5670 <infile>
1306:    48 85 c0                test    %rax,%rax
1309:    0f 84 df 00 00 00      je      13ee <main+0x115>
130f:    e8 f4 07 00 00         call    1b08 <initialize_bomb>
1314:    48 8d 3d 65 1d 00 00    lea     0x1d65(%rip),%rdi        # 3080
<_IO_stdin_used+0x80>
131b:    e8 50 fd ff ff         call    1070 <puts@plt>
1320:    48 8d 3d 99 1d 00 00    lea     0x1d99(%rip),%rdi        # 30c0
<_IO_stdin_used+0xc0>

```

```

1327:    e8 44 fd ff ff    call    1070 <puts@plt>
132c:    e8 e1 0a 00 00    call    1e12 <read_line>
1331:    48 89 c7          mov     %rax,%rdi
1334:    e8 f0 00 00 00    call    1429 <phase_1>
1339:    e8 0e 0c 00 00    call    1f4c <phase_defused>
133e:    48 8d 3d ab 1d 00 00 lea     0x1dab(%rip),%rdi =5          # 30f0
<_IO_stdin_used+0xf0>
1345:    e8 26 fd ff ff    call    1070 <puts@plt>
134a:    e8 c3 0a 00 00    call    1e12 <read_line>
134f:    48 89 c7          mov     %rax,%rdi
1352:    e8 f2 00 00 00    call    1449 <phase_2>
1357:    e8 f0 0b 00 00    call    1f4c <phase_defused>
135c:    48 8d 3d da 1c 00 00 lea     0x1cda(%rip),%rdi          # 303d
<_IO_stdin_used+0x3d>
1363:    e8 08 fd ff ff    call    1070 <puts@plt>
1368:    e8 a5 0a 00 00    call    1e12 <read_line>
136d:    48 89 c7          mov     %rax,%rdi
1370:    e8 b9 01 00 00    call    152e <phase_3>
1375:    e8 d2 0b 00 00    call    1f4c <phase_defused>
137a:    48 8d 3d cd 1c 00 00 lea     0x1ccd(%rip),%rdi          # 304e
<_IO_stdin_used+0x4e>
1381:    e8 ea fc ff ff    call    1070 <puts@plt>
1386:    e8 87 0a 00 00    call    1e12 <read_line>
138b:    48 89 c7          mov     %rax,%rdi
138e:    e8 d8 02 00 00    call    166b <phase_4>
1393:    e8 b4 0b 00 00    call    1f4c <phase_defused>
1398:    48 8d 3d 81 1d 00 00 lea     0x1d81(%rip),%rdi          # 3120
<_IO_stdin_used+0x120>
139f:    e8 cc fc ff ff    call    1070 <puts@plt>
13a4:    e8 69 0a 00 00    call    1e12 <read_line>
13a9:    48 89 c7          mov     %rax,%rdi
13ac:    e8 3a 03 00 00    call    16eb <phase_5>
13b1:    e8 96 0b 00 00    call    1f4c <phase_defused>
13b6:    48 8d 3d a0 1c 00 00 lea     0x1ca0(%rip),%rdi          # 305d
<_IO_stdin_used+0x5d>
13bd:    e8 ae fc ff ff    call    1070 <puts@plt>
13c2:    e8 4b 0a 00 00    call    1e12 <read_line>
13c7:    48 89 c7          mov     %rax,%rdi
13ca:    e8 64 03 00 00    call    1733 <phase_6>
13cf:    e8 78 0b 00 00    call    1f4c <phase_defused>

```



```

13d4:    b8 00 00 00 00      mov     $0x0,%eax
13d9:    5b                   pop     %rbx
13da:    c3                   ret
13db:    48 8b 05 6e 42 00 00  mov     0x426e(%rip),%rax      # 5650
<stdin@GLIBC_2.2.5>
13e2:    48 89 05 87 42 00 00  mov     %rax,0x4287(%rip)      # 5670 <infile>
13e9:    e9 21 ff ff ff      jmp     130f <main+0x36>
13ee:    48 8b 53 08          mov     0x8(%rbx),%rdx
13f2:    48 8b 33             mov     (%rbx),%rsi
13f5:    48 8d 3d 0a 1c 00 00  lea     0x1c0a(%rip),%rdi      # 3006
<_IO_stdin_used+0x6>
13fc:    e8 af fc ff ff      call    10b0 <printf@plt>
1401:    bf 08 00 00 00      mov     $0x8,%edi
1406:    e8 85 fd ff ff      call    1190 <exit@plt>
140b:    48 8b 36             mov     (%rsi),%rsi
140e:    48 8d 3d 0e 1c 00 00  lea     0x1c0e(%rip),%rdi      # 3023
<_IO_stdin_used+0x23>
1415:    b8 00 00 00 00      mov     $0x0,%eax
141a:    e8 91 fc ff ff      call    10b0 <printf@plt>
141f:    bf 08 00 00 00      mov     $0x8,%edi
1424:    e8 67 fd ff ff      call    1190 <exit@plt>

```

0000000000001429 <phase_1>:

```

1429:    48 83 ec 08          sub     $0x8,%rsp

```

```

142d:      48 8d 35 14 1d 00 00    lea    0x1d14(%rip),%rsi    # 3148
<_IO_stdin_used+0x148>
1434:      e8 73 06 00 00          call   1aac <strings_not_equal>
1439:      85 c0                    test   %eax,%eax
143b:      75 05                    jne    1442 <phase_1+0x19>
143d:      48 83 c4 08              add     $0x8,%rsp
1441:      c3                      ret
1442:      e8 ca 08 00 00          call   1d11 <explode_bomb>
1447:      eb f4                    jmp     143d <phase_1+0x14>

```

00000000000001449 <phase_2>:

```

1449:      55                      push   %rbp
144a:      53                      push   %rbx

```

144b: 48 83 ec 68 sub \$0x68,%rsp

函数初始化:

两个指针寄存器入栈

为栈分配空间

144f: 64 48 8b 04 25 28 00 mov %fs:0x28,%rax

1456: 00 00

1458: 48 89 44 24 58 mov %rax,0x58(%rsp)

金丝雀

145d: 31 c0 xor %eax,%eax

145f: c7 44 24 20 00 00 00 movl \$0x0,0x20(%rsp)

1466: 00

1467: c7 44 24 24 00 00 00 movl \$0x0,0x24(%rsp)

146e: 00

146f: c7 44 24 28 00 00 00 movl \$0x0,0x28(%rsp)

1476: 00

1477: c7 44 24 2c 00 00 00 movl \$0x0,0x2c(%rsp)

147e: 00

147f: c7 44 24 30 00 00 00 movl \$0x0,0x30(%rsp)

1486: 00

1487: c7 44 24 34 00 00 00 movl \$0x0,0x34(%rsp)

初始化栈变量 清理出 6 个四字节空位

//x/6d 0x20+\$rsp

148e: 00

148f: 48 b8 41 44 41 42 46 movabs \$0x4246434642414441,%rax
// BFCBAADA

1496: 43 46 42

1499: 48 ba 41 43 45 45 42 movabs \$0x46434245454341,%rdx
// FCEBECF

14a0: 43 46 00

14a3: 48 89 44 24 40 mov %rax,0x40(%rsp) //填入

14a8: 48 89 54 24 48 mov %rdx,0x48(%rsp) //填入

14ad: 48 89 e6 mov %rsp,%rsi

//载入莫名其妙的数 (神秘 15 字符串 ADABFCFBACEEBCF)

14b0: e8 1c 09 00 00 call 1dd1 <read_six_numbers> //调函数读入 input(由传参知, input 放在 rsi 里)

14b5: 83 3c 24 00 cmpl \$0x0,(%rsp) 检查输入合法性(输入非负 否则直接炸)

14b9: 78 0c js 14c7 <phase_2+0x7e>

开始操作

14bb: bb 00 00 00 00 mov \$0x0,%ebx //ebx 清零

14c0: 48 8d 6c 24 40 lea 0x40(%rsp),%rbp//rbp 指向神秘 15 字符串

主循环

14c5: eb 19 jmp 14e0 <phase_2+0x97>

14c7: e8 45 08 00 00 call 1d11 <explode_bomb>

14cc: eb ed jmp 14bb <phase_2+0x72>

14ce: 0f be 04 2b movsbl (%rbx,%rbp,1),%eax

//将%rbp 栈指针(15 串)偏移量为 rbx 的数据拷入%eax (计数器 rbp 自增引索)取出一个字母)

14d2: 83 e8 41 sub \$0x41,%eax

//0x41 对应 65 是 ASCII 的 'A'

14d5: 48 98 cltq // %eax 扩展为%rax

14d7: 83 44 84 20 01 addl \$0x1,0x20(%rsp,%rax,4) //计数

//向开头清零的那片地址写入数据了!! 基址是 0x20(%rsp)加上了%rax*4(4 的倍数)

//字符差——%rax 作为索引

14dc: 48 83 c3 01 add \$0x1,%rbx //ebx 也跟着+1

14e0: 48 89 ef mov %rbp,%rdi //传参(15 串给 rdi)

14e3: e8 a7 05 00 00 call 1a8f <string_length> //调 strlen 测 15 串长(而非 input 长)

14e8: 39 d8 cmp %ebx,%eax //strlen 默认返回到 eax

14ea: 7f e2 jg 14ce <phase_2+0x85> //eax>ebx 则跳转

//while 循环 处理完 15 个字符串后走下面 ↓

14ec: bb 00 00 00 00 mov \$0x0,%ebx //rbx 计数器清零

14f1: 48 8d 6c 24 20 lea 0x20(%rsp),%rbp//

14f6: eb 0a jmp 1502 <phase_2+0xb9>

14f8: 48 83 c3 04 add \$0x4,%rbx //从 0 反复加 4, 走几次便能满足 ==0x18

14fc: 48 83 fb 18 cmp \$0x18,%rbx

1500: 74 10 je 1512 <phase_2+0xc9> //其实是只要跳转到 14f8 就直接胜出!

1502: 8b 44 1d 00 mov 0x0(%rbp,%rbx,1),%eax //这片地址放着字母计数结果

1506: 39 04 1c cmp %eax,(%rsp,%rbx,1) //后面地址(6 块 4*int)放着输入

1509: 74 ed je 14f8 <phase_2+0xaf> //按找计数表输入数字即可进入胜利循环

150b: e8 01 08 00 00 call 1d11 <explode_bomb>

1510: eb e6 jmp 14f8 <phase_2+0xaf>

1512: 48 8b 44 24 58 mov 0x58(%rsp),%rax

1517: 64 48 2b 04 25 28 00 sub %fs:0x28,%rax

金丝雀

151e: 00 00

1520: 75 07 jne 1529 <phase_2+0xe0>

1522: 48 83 c4 68 add \$0x68,%rsp

1526: 5b pop %rbx

1527: 5d pop %rbp

```
1528:      c3                ret
1529:  e8 72 fb ff ff      call    10a0 <__stack_chk_fail@plt>
```

000000000000152e <phase_3>:

```
152e:  48 83 ec 18          sub    $0x18,%rsp
1532:  64 48 8b 04 25 28 00  mov    %fs:0x28,%rax
1539:  00 00
```

```

153b:    48 89 44 24 08        mov     %rax,0x8(%rsp)
//金丝雀
1540:    31 c0                 xor     %eax,%eax
1542:    48 8d 4c 24 04        lea     0x4(%rsp),%rcx //0x00007ff (-1)
1547:    48 89 e2             mov     %rsp,%rdx
154a:    48 8d 35 66 20 00 00  lea     0x2066(%rip),%rsi //83=>37(%d %d) 意外收获输入格式
# 35b7 <array.0+0x3d7> //由上题对 sscanf 分析知,%rsi 是传入的参数, 承载着输入格式
1551:    e8 fa fb ff ff       call    1150 <_isoc99_sscanf@plt>
1556:    83 f8 01             cmp     $0x1,%eax//可见此分支用来判断输入合法性
1559:    7e 1d             jle     1578 <phase_3+0x4a> //eax<=1 直接炸(eax 装 参数数量==2)
155b:    83 3c 24 07         cmpl    $0x7,(%rsp) // (%rsp) 存放第一个输入
155f:    0f 87 c0 00 00 00    ja     1625 <phase_3+0xf7> //rsp>7 直接炸 第一个输入<=7
1565:    8b 04 24             mov     (%rsp),%eax//传第一个输入数字
1568:    48 8d 15 51 1c 00 00  lea     0x1c51(%rip),%rdx //-65
# 31c0 <_IO_stdin_used+0x1c0>
156f:    48 63 04 82         movslq  (%rdx,%rax,4),%rax
1573:    48 01 d0             add     %rdx,%rax
1576:    ff e0             jmp     *%rax
1578:    e8 94 07 00 00     call    1d11 <explode_bomb>
157d:    eb dc             jmp     155b <phase_3+0x2d>
157f:    8b 15 8b 3b 00 00    mov     0x3b8b(%rip),%edx # 5110 <delta.1>
1585:    b8 a2 00 00 00     mov     $0xa2,%eax
158a:    29 d0             sub     %edx,%eax
158c:    8b 54 24 04         mov     0x4(%rsp),%edx//第二个输入
1590:    85 d2             test    %edx,%edx
1592:    78 04             js     1598 <phase_3+0x6a> //edx<0 炸 故第二个数>=0
1594:    39 c2             cmp     %eax,%edx//eax 是 231
1596:    74 05             je     159d <phase_3+0x6f> //此处 eax 必须等于第二个数, 不然直接炸
1598:    e8 74 07 00 00     call    1d11 <explode_bomb> //故第二个数必须是 231
159d:    48 8b 44 24 08     mov     0x8(%rsp),%rax
15a2:    64 48 2b 04 25 28 00  sub     %fs:0x28,%rax
15a9:    00 00
15ab:    0f 85 83 00 00 00    jne     1634 <phase_3+0x106> //指向<__stack_chk_fail@plt>
//金丝雀
15b1:    48 83 c4 18         add     $0x18,%rsp
15b5:    c3                 ret
15b6:    8b 15 54 3b 00 00    mov     0x3b54(%rip),%edx # 5110 <delta.1>
15bc:    b8 fb 00 00 00     mov     $0xfb,%eax
15c1:    29 d0             sub     %edx,%eax

```

```

15c3:    eb c7                jmp     158c <phase_3+0x5e>
15c5:    8b 15 45 3b 00 00    mov     0x3b45(%rip),%edx        # 5110 <delta.1>
15cb:    b8 4c 02 00 00      mov     $0x24c,%eax
15d0:    29 d0                sub     %edx,%eax
15d2:    eb b8                jmp     158c <phase_3+0x5e>
15d4:    8b 15 36 3b 00 00    mov     0x3b36(%rip),%edx        # 5110 <delta.1>
15da:    b8 3c 00 00 00      mov     $0x3c,%eax
15df:    29 d0                sub     %edx,%eax
15e1:    eb a9                jmp     158c <phase_3+0x5e>
15e3:    8b 15 27 3b 00 00    mov     0x3b27(%rip),%edx        # 5110 <delta.1>
15e9:    b8 e2 02 00 00      mov     $0x2e2,%eax
15ee:    29 d0                sub     %edx,%eax
15f0:    eb 9a                jmp     158c <phase_3+0x5e>
15f2:    8b 15 18 3b 00 00    mov     0x3b18(%rip),%edx        # 5110 <delta.1>
15f8:    b8 2f 02 00 00      mov     $0x22f,%eax
15fd:    29 d0                sub     %edx,%eax
15ff:    eb 8b                jmp     158c <phase_3+0x5e>
1601:    8b 15 09 3b 00 00    mov     0x3b09(%rip),%edx        # 5110 <delta.1>
1607:    b8 10 03 00 00      mov     $0x310,%eax
160c:    29 d0                sub     %edx,%eax
160e:    e9 79 ff ff ff      jmp     158c <phase_3+0x5e> //中间好大一段没用上
1613:    8b 15 f7 3a 00 00    mov     0x3af7(%rip),%edx //0x2d4
# 5110 <delta.1> //rip== -72
1619:    b8 bb 03 00 00      mov     $0x3bb,%eax //0x3bb
161e:    29 d0                sub     %edx,%eax //0xe7 做差 231(十进制)or-25
1620:    e9 67 ff ff ff      jmp     158c <phase_3+0x5e>
1625:    e8 e7 06 00 00      call    1d11 <explode_bomb>
162a:    b8 00 00 00 00      mov     $0x0,%eax
162f:    e9 58 ff ff ff      jmp     158c <phase_3+0x5e>
1634:    e8 67 fa ff ff      call    10a0 <__stack_chk_fail@plt>

```

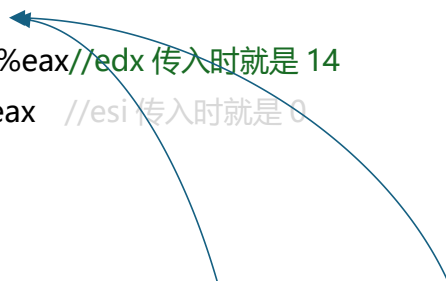
1000

00000000000001639 <func4>:(edi=>first input)

```

1639:    53                push    %rbx
163a:    89 d0                mov     %edx,%eax //edx 传入时就是 14
163c:    29 f0                sub     %esi,%eax //esi 传入时就是 0

```



163e:	89 c3	mov	%eax,%ebx//==14
1640:	c1 eb 1f	shr	\$0x1f,%ebx //将 14 逻辑右移 31 位(取符号位)=>0
1643:	01 c3	add	%eax,%ebx
1645:	d1 fb	sar	%ebx//将 14 算术逻辑右移 1 位==7
1647:	01 f3	add	%esi,%ebx
1649:	39 fb	cmp	%edi,%ebx
164b:	7f 06	jg	1653 <func4+0x1a> // 第一个数<7
164d:	7c 10	jl	165f <func4+0x26> // 第一个数>7
164f:	89 d8	mov	%ebx,%eax
1651:	5b	pop	%rbx
1652:	c3	ret	
1653:	8d 53 ff	lea	-0x1(%rbx),%edx//edx=6
1656:	e8 de ff ff ff	call	1639 <func4> //改换初始值, 重新执行一遍函数
165b:	01 c3	add	%eax,%ebx
165d:	eb f0	jmp	164f <func4+0x16>
165f:	8d 73 01	lea	0x1(%rbx),%esi //esi==8
1662:	e8 d2 ff ff ff	call	1639 <func4>
1667:	01 c3	add	%eax,%ebx
1669:	eb e4	jmp	164f <func4+0x16>

二分查找? !

000000000000166b <phase_4>:

```
166b:      53                push   %rbx
166c:    48 83 ec 10       sub    $0x10,%rsp
1670:    64 48 8b 04 25 28 00 mov    %fs:0x28,%rax
1677:      00 00
1679:    48 89 44 24 08     mov    %rax,0x8(%rsp)
```

//金丝雀

```
167e:    31 c0             xor    %eax,%eax
1680:    48 8d 4c 24 04     lea    0x4(%rsp),%rcx //-1
1685:    48 89 e2           mov    %rsp,%rdx //56
1688:    48 8d 35 28 1f 00 00 lea     0x1f28(%rip),%rsi    # 35b7
```

```
(gdb) x/s 0x1f28+$rip
0x5555555575b7: "%d %d"
```

<array.0+0x3d7> //还是老样子，进 sscanf 前传入 input 格式

```
168f:    e8 bc fa ff ff     call   1150 <__isoc99_sscanf@plt>
1694:    83 f8 02           cmp    $0x2,%eax //按格式输,rax 就一定是 2
1697:    75 06             jne    169f <phase_4+0x34>
1699:    83 3c 24 0e       cmpl   $0xe,(%rsp) //第一个数要求<=14
169d:    76 05             jbe    16a4 <phase_4+0x39>
169f:    e8 6d 06 00 00     call   1d11 <explode_bomb>
16a4:    8b 44 24 04       mov    0x4(%rsp),%eax //第二个数传入 eax
16a8:    8d 58 f4          lea    -0xc(%rax),%ebx
16ab:    89 5c 24 04       mov    %ebx,0x4(%rsp) //第二个数 -=12
16af:    ba 0e 00 00 00     mov    $0xe,%edx //edx 置 14
16b4:    be 00 00 00 00     mov    $0x0,%esi//esi 置 0
16b9:    8b 3c 24          mov    (%rsp),%edi//第一个数传入 edi
16bc:    e8 78 ff ff ff     call   1639 <func4> //传入第一个数，传出 ebx(第二个数-12),eax
```

//func4 蕴含着一个超级恶心复杂的映射，在里面貌似有个二分查找猜传入的第一个数，猜中后映射成新的 eax 传出 因为第一个数<=14 所以直接试验出 func4 的映射表比每种情况挨个走汇编快捷高效，不妨直接试！

14=>45

13=>31

12=>43

11=>11

10=>37

9=>27

8=>35

7=>7

6=>21

5=>15

4=>19 !!!!! 终于打表试出来了

16c1:	83 fb 13	cmp	\$0x13,%ebx// ebx==19 才不爆炸=> 第二必须是 31
16c4:	75 05	jne	16cb <phase_4+0x60>
16c6:	83 f8 13	cmp	\$0x13,%eax//eax==19 才不爆炸
16c9:	74 05	je	16d0 <phase_4+0x65>
16cb:	e8 41 06 00 00	call	1d11 <explode_bomb>
16d0:	48 8b 44 24 08	mov	0x8(%rsp),%rax
16d5:	64 48 2b 04 25 28 00	sub	%fs:0x28,%rax
16dc:	00 00		
16de:	75 06	jne	16e6 <phase_4+0x7b>
16e0:	48 83 c4 10	add	\$0x10,%rsp
16e4:	5b	pop	%rbx
16e5:	c3	ret	
16e6:	e8 b5 f9 ff ff	call	10a0 <__stack_chk_fail@plt>

00000000000016eb <phase_5>:

```
16eb:    53                push    %rbx
16ec:    48 89 fb          mov     %rdi,%rbx
16ef:    e8 9b 03 00 00    call    1a8f <string_length>
16f4:    83 f8 06          cmp     $0x6,%eax
16f7:    75 2c             jne     1725 <phase_5+0x3a> //输入字符长!=6 就炸
16f9:    48 89 d8          mov     %rbx,%rax
16fc:    48 8d 7b 06       lea     0x6(%rbx),%rdi
1700:    b9 00 00 00 00    mov     $0x0,%ecx
1705:    48 8d 35 d4 1a 00 00 lea     0x1ad4(%rip),%rsi //数组[0]传入 rsi      # 31e0
```

<array.0>

//遇到一个奇怪数组，把他全打印出来（不知大小超量打印即可，越界访问会有明显提示）

```
<array.0>:    2    0    0    0    10    0    0    0
<array.0+8>:    6    0    0    0    1    0    0    0
<array.0+16>:  12    0    0    0    16    0    0    0
<array.0+24>:    9    0    0    0    3    0    0    0
<array.0+32>:    4    0    0    0    7    0    0    0
<array.0+40>:   14    0    0    0    5    0    0    0
<array.0+48>:   11    0    0    0    8    0    0    0
<array.0+56>:   15    0    0    0   13    0    0    0
```

```
170c:    0f b6 10          movzbl (%rax),%edx //第一字符的 ascii
170f:    83 e2 0f          and     $0xf,%edx //& 1111
1712:    03 0c 96          add     (%rsi,%rdx,4),%ecx
1715:    48 83 c0 01       add     $0x1,%rax
1719:    48 39 f8          cmp     %rdi,%rax
171c:    75 ee             jne     170c <phase_5+0x21>
171e:    83 f9 1f          cmp     $0x1f,%ecx
1721:    75 09             jne     172c <phase_5+0x41>
1723:    5b               pop     %rbx
1724:    c3               ret
1725:    e8 e7 05 00 00    call    1d11 <explode_bomb>
172a:    eb cd             jmp     16f9 <phase_5+0xe>
172c:    e8 e0 05 00 00    call    1d11 <explode_bomb>
1731:    eb f0             jmp     1723 <phase_5+0x38>
```

0000000000001733 <phase_6>:

```
1733: 41 56      push    %r14
1735: 41 55      push    %r13
1737: 41 54      push    %r12
1739: 55         push    %rbp
173a: 53         push    %rbx
173b: 48 83 ec 60 sub     $0x60,%rsp
173f: 64 48 8b 04 25 28 00 mov     %fs:0x28,%rax
1746: 00 00
1748: 48 89 44 24 58 mov     %rax,0x58(%rsp)
```

//金丝雀

```
174d: 31 c0      xor     %eax,%eax
174f: 49 89 e5    mov     %rsp,%r13 //输入地址传给 r13
1752: 4c 89 ee    mov     %r13,%rsi //输入地址传给 rsi 为 read_6_num 准备
1755: e8 77 06 00 00 call    1dd1 <read_six_numbers> //老朋友了...
175a: 41 be 01 00 00 00 mov     $0x1,%r14d //r14d=1
1760: 49 89 e4    mov     %rsp,%r12 //输入地址传给 r12
1763: eb 28      jmp     178d <phase_6+0x5a>
1765: e8 a7 05 00 00 call    1d11 <explode_bomb>
176a: eb 30      jmp     179c <phase_6+0x69>
176c: 48 83 c3 01 add     $0x1,%rbx
1770: 83 fb 05    cmp     $0x5,%ebx //注1: 引索到 6, 即扫描完就跳出循环
1773: 7f 10      jg      1785 <phase_6+0x52> //ebx>5 就跳转
1775: 41 8b 04 9c mov     (%r12,%rbx,4),%eax //读下一个数
1779: 39 45 00    cmp     %eax,0x0(%rbp) //与第一个数相比
177c: 75 ee      jne     176c <phase_6+0x39> //输入数不能与前驱重复, 否则炸
177e: e8 8e 05 00 00 call    1d11 <explode_bomb>
1783: eb e7      jmp     176c <phase_6+0x39>
1785: 49 83 c3 01 add     $0x1,%r14 //r14+1=2 r14作为input 引索
1789: 49 83 c3 01 add     $0x4,%r13 //输入地址指针+4
178d: 4c 89 ee    mov     %r13,%rbp //输入地址传给 rbp
1790: 41 8b 45 00 mov     0x0(%r13),%eax //一个input 数传给 eax
1794: 83 e8 01    sub     $0x1,%eax //input -1
1797: 83 f8 05    cmp     $0x5,%eax //eax-1>5 炸 //每个数都写<=6
179a: 77 c9      ja      1765 <phase_6+0x32> //ja 无符号大于 故0不行
179c: 41 83 fe 05 cmp     $0x5,%r14d //引索 r14d >5 则跳转
17a0: 7f 05      jg      17a7 <phase_6+0x74> //满足全员<=6 跳出循环
17a2: 4c 89 f3    mov     %r14,%rbx //rbx 接受 r14 (1) rbx 作为input 数组引索
17a5: eb ce      jmp     1775 <phase_6+0x42>
```

考察输入范围
缩小 1(第一个
input 毕业)

查重循环

查>6 循环

17a7: be 00 00 00 00 mov \$0x0,%esi //esi 作为 input 数组索引,此处归 0
 17ac: 8b 0c b4 mov (%rsp,%rsi,4),%ecx //入读一个 input 存入 ecx
 17af: b8 01 00 00 00 mov \$0x1,%eax
 17b4: 48 8d 15 25 3a 00 00 lea 0x3a25(%rip),%rdx //再执行时, 已换了串链表

(gdb) x/40d \$rdx

```
0x5555555591e0 <node1>: 546 1
0x5555555591f0 <node2>: 419 2
0x555555559200 <node3>: 858 3
0x555555559210 <node4>: 68 4
0x555555559220 <node5>: 497 5
0x555555559130 <node6>: 745 6
```

51e0 <node1>

//遇到一个奇怪链表, 把他全打印出来 (不知大小超量打印即可, 越界访问会有明显提示)

黑话: input 6 个数代号:

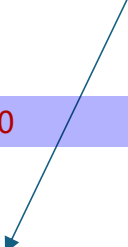
A B C D E F

17bb: 83 f9 01 cmp \$0x1,%ecx //有输入 <=1 就跳转
 17be: 7e 0b jle 17cb <phase_6+0x98>
 17c0: 48 8b 52 08 mov 0x8(%rdx),%rdx //走入下一个结点(node
 17c4: 83 c0 01 add \$0x1,%eax
 17c7: 39 c8 cmp %ecx,%eax //直到找到输入对应的索引
 17c9: 75 f5 jne 17c0 <phase_6+0x8d>
 17cb: 48 89 54 f4 20 mov %rdx,0x20(%rsp,%rsi,8) //相应结点值传入 input 地址大后方
 17d0: 48 83 c6 01 add \$0x1,%rsi
 17d4: 48 83 f cmp \$0x6,%rsi //input 索引不到 6 就一直跳转(循环)
 17d8: 75 d2 jne 17ac <phase_6+0x79>
 17da: 48 8b 5 mov 0x20(%rsp),%rbx //rbx=> 1nodeA
 17df: 48 8b 4 mov 0x28(%rsp),%rax //rax=> 1nodeB
 17e4: 48 89 4 mov %rax,0x8(%rbx) //2node5+8=1431671312
 17e8: 48 8b 5 mov 0x30(%rsp),%rdx //rdx=> 2node2 =-93
 17ed: 48 89 5 mov %rdx,0x8(%rax) //1node4+8 => (32=>-16)
 17f1: 48 8b 4 mov 0x38(%rsp),%rax //rax:1node4=> 2node1=34
 17f6: 48 89 4 mov %rax,0x8(%rdx) //2node2+8=-32
 17fa: 48 8b 5 mov 0x40(%rsp),%rdx //rdx=> 2node6 =-23
 17ff: 48 89 50 mov %rdx,0x8(%rax) //2node1+8=48
 1803: 48 8b 4 mov 0x48(%rsp),%rax //2node3=90
 1808: 48 89 4 mov %rax,0x8(%rdx) //2node6+8=0
 180c: 48 c7 40 00 movq \$0x0,0x8(%rax) //2node3+8=0
 1813: 00
 1814: bd 05 00 00 00 mov \$0x5,%ebp
 1819: eb 09 jmp 1824 <phase_6+0xf1>
 181b: 48 8b 5b 08 mov 0x8(%rbx),%rbx
 181f: 83 ed 01 sub \$0x1,%ebp
 1822: 74 11 je 1835 <phase_6+0x102>
 1824: 48 8b 43 08 mov 0x8(%rbx),%rax //1node4=68
 1828: 8b 00 mov (%rax),%eax //eax=68
 182a: 39 03 cmp %eax,(%rbx) //2nodeA <= 1nodeB
 182c: 7e ed jle 181b <phase_6+0xe8>

链表
乱七
八糟
的连接
操作
(红色
注释
是第一
次
啥也
不懂
时写
的, 不
必理
会)

查升序循环

182e:	e8 de 04 00 00	call	1d11 <explode_bomb>
1833:	eb e6	jmp	181b <phase_6+0xe8>
1835:	48 8b 44 24 58	mov	0x58(%rsp),%rax
183a:	64 48 2b 04 25 28 00	sub	%fs:0x28,%rax
1841:	00 00		
1843:	75 0d	jne	1852 <phase_6+0x11f>
1845:	48 83 c4 60	add	\$0x60,%rsp
1849:	5b	pop	%rbx
184a:	5d	pop	%rbp
184b:	41 5c	pop	%r12
184d:	41 5d	pop	%r13
184f:	41 5e	pop	%r14
1851:	c3	ret	
1852:	e8 49 f8 ff ff	call	10a0 <__stack_chk_fail@plt>



0000000000001857 <fun7>:

1857:	48 83 ec 38	sub	\$0x38,%rsp
185b:	41 89	mov	%edx,%r9d
185e:	64 48	mov	%fs:0x28,%rax
1865:	00 00		
1867:	48 89	mov	%rax,0x28(%rsp)
186c:	31 c0	xor	%eax,%eax
186e:	c7 04 24 00 00 00 00	movl	\$0x0,(%rsp)
1875:	c7 44 24 04 00 00 00	movl	\$0x0,0x4(%rsp)
187c:	00		
187d:	c7 44 24 08 01 00 00	movl	\$0x1,0x8(%rsp)
1884:	00		
1885:	c7 44 24 0c ff ff ff	movl	\$0xffffffff,0xc(%rsp)
188c:	ff		
188d:	c7 44 24 10 01 00 00	movl	\$0x1,0x10(%rsp)
1894:	00		
1895:	c7 44 24 14 ff ff ff	movl	\$0xffffffff,0x14(%rsp)
189c:	ff		
189d:	c7 44 24 18 00 00 00	movl	\$0x0,0x18(%rsp)
18a4:	00		
18a5:	c7 44 24 1c 00 00 00	movl	\$0x0,0x1c(%rsp)
18ac:	00	//Local [8] : rsp: 0 0 1 -1 1 -1 0 0	
18ad:	8d 56 07	lea	0x7(%rsi),%edx
18b0:	85 f6	test	%esi,%esi
18b2:	0f 49 d6	cmovns	%esi,%edx //esi 非负 则 传入 edx
18b5:	c1 fa 03	sar	\$0x3,%edx // 算数右移 3 位
18b8:	89 f0	mov	%esi,%eax
18ba:	c1 fb	sar	\$0x1f,%eax //esi 算数右移 31 位 (取符号位)
18bd:	c1 fd	shr	\$0x1d,%eax//esi 逻辑右移 29 位
18c0:	01 c6	add	%eax,%esi
18c2:	83 e6 07	and	\$0x7,%esi //esi 取余 8
18c5:	29 c6	sub	%eax,%esi
18c7:	39 f2	cmp	%esi,%edx
18c9:	75 1b	jne	18e6 <fun7+0x8f>
18cb:	83 fa 07	cmp	\$0x7,%edx
18ce:	75 16	jne	18e6 <fun7+0x8f> // edx = 7 且 edx=esi
18d0:	b8 00 00 00 00	mov	\$0x0,%eax
18d5:	41 83 f9 04	cmp	\$0x4,%r9d //r9d==4

初始:

Edx=0

Esi =0

神秘值写入数组

rax,esi 的一系列映射计算

Local [8]: 0 0 1 -1 1 -1 0 0

r8d 增值表:

	rax%4	r8d+ ?
18db:		
18de:		
18e1:		
18e4:		
18e6:	0	1
18e9:	1	-1
18ec:		
18ef:	2	1
18f3:		
18f6:	3	-1
18f9:		
18fe:		
1902:		
1905:		
1909:		
1912:		
1914:		
1918:		
1919:		
1920:		
1922:		
1924:		
1928:		
192b:		
192d:		
192f:		
1932:		
1937:		
193c:		
1940:		
1942:		
1946:		
1949:		
194d:		
194f:		
1953:		
1956:		
1959:		
195e:		

jne 1904 <fun7+0xad>

cmpb \$0x0,(%rcx) //rcx==0 则设 al=1

sete %al //根据 ZF 标志位的值设置一个变量的值是 1 和 0

movzbl %al,%eax

jmp 1904 <fun7+0xad>

movzbl (%rcx),%eax //input_strings 的 ascii 码

and \$0x3,%eax //eax 取余 4

add (%rsp,%rax,4),%edx edx+=Local[eax]

add 0x10(%rsp,%rax,4),%esi esi+= Local[eax+4]

mov %edx,%r8d //r8d = edx

or %esi,%r8d //r8d=r8d | esi

mov \$0x0,%eax

cmp \$0x7,%r8d

jbe 1919 <fun7+0xc2> // r8d 必须 [0,7]

mov 0x28(%rsp),%rdx

sub %fs:0x28,%rdx

jne 1975 <fun7+0x11e>

add \$0x38,%rsp

ret

lea 0x3850(%rip),%r8

5170 <line1>

test %edx,%edx

jle 192f <fun7+0xd8> //edx<=0 就跳转

mov 0x8(%r8),%r8

add \$0x1,%eax

cmp %eax,%edx

jne 1924 <fun7+0xcd>

movslq %esi,%rax

movzbl (%r8,%rax,1),%r8d

mov \$0x0,%eax

cmp \$0x1,%r8b //r8b==1 才能走出循环

je 1904 <fun7+0xad> //从这里走出循环不行 (eax=0)

movsbl %r8b,%r8d

movslq %r9d,%rax

cmp (%rdi,%rax,4),%r8d // rdi 参数指向最开始的数组

je 1960 <fun7+0x109>

add \$0x1,%rcx // rcx 参数指向 input 里 向后再读一数

lea (%rsi,%rdx,8),%esi

mov %r9d,%edx

call 1857 <fun7> //典型递归结构

jmp 1904 <fun7+0xad>

1960: 48 83 c1 01
1964: 41 83 c1 01
1968: 8d 34 d6
196b: 44 89 ca
196e: e8 e4 fe ff ff
1973: eb 8f
1975: e8 26 f7 ff ff

```
add    $0x1,%rcx  
add    $0x1,%r9d  
lea    (%rsi,%rdx,8),%esi  
mov    %r9d,%edx
```

call 1857 <fun7> //典型递归结构

jmp 1904 <fun7+0xad>

call 10a0 <__stack_chk_fail@plt>

000000000000197a <secret_phase>:

197a:	53	push	%rbx	
197b:	48 83 ec 20	sub	\$0x20,%rsp	
197f:	64 48 8b 04 25 28 00	mov	%fs:0x28,%rax	
1986:	00 00			
1988:	48 89 44 24 18	mov	%rax,0x18(%rsp)	
198d:	31 c0	xor	%eax,%eax	
198f:	c7 04 24 04 00 00 00	movl	\$0x4,(%rsp)	
1996:	c7 44 24 04 03 00 00	movl	\$0x3,0x4(%rsp)	
199d:	00			
199e:	c7 44 24 08 02 00 00	movl	\$0x2,0x8(%rsp)	
19a5:	00			
19a6:	c7 44 24 0c 05 00 00	movl	\$0x5,0xc(%rsp)	
19ad:	00			//录入数组 rsp 4 3 2 5
19ae:	e8 5f 04 00 00	call	1e12 <read_line>	//读取第七行输入
19b3:	48 89 c3	mov	%rax,%rbx	
19b6:	48 89 c7	mov	%rax,%rdi	
19b9:	e8 d1 00 00 00	call	1a8f <string_length>	
19be:	83 f8 46	cmp	\$0x46,%eax	
19c1:	7f 40	jg	1a03 <secret_phase+0x89>	//输入长度大于 70 就炸
19c3:	48 89 e7	mov	<u>%rsp,%rdi</u>	//input 传参给 rdi
19c6:	48 89 d9	mov	%rbx,%rcx	
19c9:	ba 00 00 00 00	mov	\$0x0,%edx	
19ce:	be 00 00 00 00	mov	\$0x0,%esi	
19d3:	e8 7f fe ff ff	call	1857 <fun7>	
19d8:	85 c0	test	%eax,%eax	
19da:	74 2e	je	1a0a <secret_phase+0x90>	//eax==0 就炸

//由此可见 fun7 返回值坚决不能为 0

19dc:	48 8d 3d 95 17 00 00	lea	0x1795(%rip),%rdi	# 3178
<_IO_stdin_used+0x178>				
19e3:	e8 88 f6 ff ff	call	1070 <puts@plt>	
19e8:	e8 5f 05 00 00	call	1f4c <phase_defused>	
19ed:	48 8b 44 24 18	mov	0x18(%rsp),%rax	
19f2:	64 48 2b 04 25 28 00	sub	%fs:0x28,%rax	
19f9:	00 00			
19fb:	75 14	jne	1a11 <secret_phase+0x97>	
19fd:	48 83 c4 20	add	\$0x20,%rsp	
1a01:	5b	pop	%rbx	

```

1a02:      c3                ret
1a03:      e8 09 03 00 00      call 1d11 <explode_bomb>
1a08:      eb b9                jmp 19c3 <secret_phase+0x49>
1a0a:      e8 02 03 00 00      call 1d11 <explode_bomb>
1a0f:      eb cb                jmp 19dc <secret_phase+0x62>
1a11:      e8 8a f6 ff ff      call 10a0 <__stack_chk_fail@plt>

```

00000000000001a16 <sig_handler>:

```

1a16:      48 83 ec 08          sub $0x8,%rsp
1a1a:      48 8d 3d ff 17 00 00  lea 0x17ff(%rip),%rdi    # 3220

```

<array.0+0x40>

```

1a21:      e8 4a f6 ff ff      call 1070 <puts@plt>
1a26:      bf 03 00 00 00      mov $0x3,%edi
1a2b:      e8 80 f7 ff ff      call 11b0 <sleep@plt>
1a30:      48 8d 3d a2 19 00 00  lea 0x19a2(%rip),%rdi    # 33d9

```

<array.0+0x1f9>

```

1a37:      b8 00 00 00 00      mov $0x0,%eax
1a3c:      e8 6f f6 ff ff      call 10b0 <printf@plt>
1a41:      48 8b 3d f8 3b 00 00  mov 0x3bf8(%rip),%rdi    # 5640

```

<stdout@GLIBC_2.2.5>

```

1a48:      e8 f3 f6 ff ff      call 1140 <fflush@plt>
1a4d:      bf 01 00 00 00      mov $0x1,%edi
1a52:      e8 59 f7 ff ff      call 11b0 <sleep@plt>
1a57:      48 8d 3d 83 19 00 00  lea 0x1983(%rip),%rdi    # 33e1

```

<array.0+0x201>

```

1a5e:      e8 0d f6 ff ff      call 1070 <puts@plt>
1a63:      bf 10 00 00 00      mov $0x10,%edi
1a68:      e8 23 f7 ff ff      call 1190 <exit@plt>

```

0000000000001a6d <invalid_phase>:

```
1a6d: 48 83 ec 08      sub    $0x8,%rsp
1a71: 48 89 fe         mov    %rdi,%rsi
1a74: 48 8d 3d 70 19 00 00 lea    0x1970(%rip),%rdi    # 33eb
<array.0+0x20b>
1a7b: b8 00 00 00 00   mov    $0x0,%eax
1a80: e8 2b f6 ff ff   call   10b0 <printf@plt>
1a85: bf 08 00 00 00   mov    $0x8,%edi
1a8a: e8 01 f7 ff ff   call   1190 <exit@plt>
```

0000000000001a8f <string_length>:

```
1a8f: 80 3f 00         cmpb   $0x0,(%rdi)
1a92: 74 12            je     1aa6 <string_length+0x17>
1a94: b8 00 00 00 00   mov    $0x0,%eax
1a99: 48 83 c7 01      add    $0x1,%rdi
1a9d: 83 c0 01         add    $0x1,%eax
1aa0: 80 3f 00         cmpb   $0x0,(%rdi)
1aa3: 75 f4            jne    1a99 <string_length+0xa>
1aa5: c3              ret
1aa6: b8 00 00 00 00   mov    $0x0,%eax
1aab: c3              ret
```

0000000000001aac <strings_not_equal>:

```
1aac: 41 54           push   %r12
1aae: 55             push   %rbp
1aaf: 53             push   %rbx
1ab0: 48 89 fb       mov    %rdi,%rbx
1ab3: 48 89 f5       mov    %rsi,%rbp
1ab6: e8 d4 ff ff ff call   1a8f <string_length>
1abb: 41 89 c4       mov    %eax,%r12d
1abe: 48 89 ef       mov    %rbp,%rdi
1ac1: e8 c9 ff ff ff call   1a8f <string_length>
1ac6: 89 c2         mov    %eax,%edx
1ac8: b8 01 00 00 00 mov    $0x1,%eax
1acd: 41 39 d4       cmp    %edx,%r12d
1ad0: 75 31         jne    1b03 <strings_not_equal+0x57>
1ad2: 0f b6 13      movzbl (%rbx),%edx
1ad5: 84 d2         test   %dl,%dl
1ad7: 74 1e         je     1af7 <strings_not_equal+0x4b>
1ad9: b8 00 00 00 00 mov    $0x0,%eax
```

1ade:	38 54 05 00	cmp	%dl,0x0(%rbp,%rax,1)
1ae2:	75 1a	jne	1afe <strings_not_equal+0x52>
1ae4:	48 83 c0 01	add	\$0x1,%rax
1ae8:	0f b6 14 03	movzbl	(%rbx,%rax,1),%edx
1aec:	84 d2	test	%dl,%dl
1aee:	75 ee	jne	1ade <strings_not_equal+0x32>
1af0:	b8 00 00 00 00	mov	\$0x0,%eax
1af5:	eb 0c	jmp	1b03 <strings_not_equal+0x57>
1af7:	b8 00 00 00 00	mov	\$0x0,%eax
1afc:	eb 05	jmp	1b03 <strings_not_equal+0x57>
1afe:	b8 01 00 00 00	mov	\$0x1,%eax
1b03:	5b	pop	%rbx
1b04:	5d	pop	%rbp
1b05:	41 5c	pop	%r12
1b07:	c3	ret	

0000000000001b08 <initialize_bomb>:

```
1b08:      55                push   %rbp
1b09:      53                push   %rbx
1b0a:    48 83 ec 58        sub    $0x58,%rsp
1b0e:    64 48 8b 04 25 28 00 mov    %fs:0x28,%rax
1b15:    00 00
1b17:    48 89 44 24 48        mov    %rax,0x48(%rsp)
1b1c:    31 c0                xor     %eax,%eax
1b1e:    48 8d 35 f1 fe ff ff lea     -0x10f(%rip),%rsi    # 1a16 <sig_handler>
1b25:    bf 02 00 00 00        mov    $0x2,%edi
1b2a:    e8 e1 f5 ff ff        call   1110 <signal@plt>
1b2f:    48 8b 3d 0a 37 00 00 mov     0x370a(%rip),%rdi    # 5240
```

<host_table>

```
1b36:    48 85 ff            test   %rdi,%rdi
1b39:    74 23                je     1b5e <initialize_bomb+0x56>
1b3b:    48 8d 1d 06 37 00 00 lea     0x3706(%rip),%rbx    # 5248
```

<host_table+0x8>

```
1b42:    48 89 e5            mov    %rsp,%rbp
1b45:    48 89 ee            mov    %rbp,%rsi
1b48:    e8 f3 f4 ff ff        call   1040 <strcasecmp@plt>
1b4d:    85 c0                test   %eax,%eax
1b4f:    74 0d                je     1b5e <initialize_bomb+0x56>
1b51:    48 83 c3 08          add     $0x8,%rbx
1b55:    48 8b 7b f8          mov     -0x8(%rbx),%rdi
1b59:    48 85 ff            test   %rdi,%rdi
1b5c:    75 e7                jne    1b45 <initialize_bomb+0x3d>
1b5e:    48 8d 3d 97 18 00 00 lea     0x1897(%rip),%rdi    # 33fc
```

<array.0+0x21c>

```
1b65:    e8 06 f5 ff ff        call   1070 <puts@plt>
1b6a:    48 8d 3d 97 18 00 00 lea     0x1897(%rip),%rdi    # 3408
```

<array.0+0x228>

```
1b71:    e8 fa f4 ff ff        call   1070 <puts@plt>
1b76:    48 8d 3d 97 18 00 00 lea     0x1897(%rip),%rdi    # 3414
```

<array.0+0x234>

```
1b7d:    e8 ee f4 ff ff        call   1070 <puts@plt>
1b82:    48 8d 3d 97 18 00 00 lea     0x1897(%rip),%rdi    # 3420
```

<array.0+0x240>

```
1b89:    e8 e2 f4 ff ff        call   1070 <puts@plt>
1b8e:    48 8d 3d 97 18 00 00 lea     0x1897(%rip),%rdi    # 342c
```

<array.0+0x24c>

```

1b95:    e8 d6 f4 ff ff      call   1070 <puts@plt>
1b9a:    48 8b 44 24 48      mov     0x48(%rsp),%rax
1b9f:    64 48 2b 04 25 28 00 sub     %fs:0x28,%rax
1ba6:    00 00
1ba8:    75 07               jne     1bb1 <initialize_bomb+0xa9>
1baa:    48 83 c4 58         add     $0x58,%rsp
1bae:    5b                  pop     %rbx
1baf:    5d                  pop     %rbp
1bb0:    c3                  ret
1bb1:    e8 ea f4 ff ff      call   10a0 <__stack_chk_fail@plt>

```

0000000000001bb6 <initialize_bomb_solve>:

```

1bb6:    c3                  ret

```

0000000000001bb7 <blank_line>:

```

1bb7:    55                  push    %rbp
1bb8:    53                  push    %rbx
1bb9:    48 83 ec 08         sub     $0x8,%rsp
1bbd:    48 89 fd            mov     %rdi,%rbp
1bc0:    0f b6 5d 00         movzbl 0x0(%rbp),%ebx
1bc4:    84 db              test    %bl,%bl
1bc6:    74 1e              je      1be6 <blank_line+0x2f>
1bc8:    e8 f3 f5 ff ff      call   11c0 <__ctype_b_loc@plt>
1bcd:    48 83 c5 01         add     $0x1,%rbp
1bd1:    48 0f be db         movsbq  %bl,%rbx
1bd5:    48 8b 00            mov     (%rax),%rax
1bd8:    f6 44 58 01 20      testb  $0x20,0x1(%rax,%rbx,2)
1bdd:    75 e1              jne     1bc0 <blank_line+0x9>
1bdf:    b8 00 00 00 00      mov     $0x0,%eax
1be4:    eb 05              jmp     1beb <blank_line+0x34>
1be6:    b8 01 00 00 00      mov     $0x1,%eax
1beb:    48 83 c4 08         add     $0x8,%rsp
1bef:    5b                  pop     %rbx
1bf0:    5d                  pop     %rbp
1bf1:    c3                  ret

```

0000000000001bf2 <skip>:

```

1bf2:    55                  push    %rbp
1bf3:    53                  push    %rbx
1bf4:    48 83 ec 08         sub     $0x8,%rsp

```

```

1bf8: 48 8d 2d 01 3b 00 00 lea 0x3b01(%rip),%rbp # 5700
<input_strings>
1bff: 48 63 15 f2 3a 00 00 movslq 0x3af2(%rip),%rdx # 56f8
<num_input_strings>
1c06: 48 89 d0 mov %rdx,%rax
1c09: 48 c1 e0 04 shl $0x4,%rax
1c0d: 48 29 d0 sub %rdx,%rax
1c10: 48 8d 7c c5 00 lea 0x0(%rbp,%rax,8),%rdi
1c15: 48 8b 15 54 3a 00 00 mov 0x3a54(%rip),%rdx # 5670 <infile>
1c1c: be 78 00 00 00 mov $0x78,%esi
1c21: e8 ca f4 ff ff call 10f0 <fgets@plt> //读取了最后一行
1c26: 48 89 c3 mov %rax,%rbx
1c29: 48 85 c0 test %rax,%rax
1c2c: 74 0c je 1c3a <skip+0x48>
1c2e: 48 89 c7 mov %rax,%rdi
1c31: e8 81 ff ff ff call 1bb7 <blank_line>
1c36: 85 c0 test %eax,%eax
1c38: 75 c5 jne 1bff <skip+0xd>
1c3a: 48 89 d8 mov %rbx,%rax
1c3d: 48 83 c4 08 add $0x8,%rsp
1c41: 5b pop %rbx
1c42: 5d pop %rbp
1c43: c3 ret

```

0x01 test 和 je 连用

```

1 TEST EAX,EAX
2 JE some_address

```

这段代码的含义是：

如果eax==0的话就跳转到“some_address”。

0000000000001c44 <send_msg>:

```

1c44: 53 push %rbx
1c45: 48 81 ec 10 40 00 00 sub $0x4010,%rsp
1c4c: 64 48 8b 04 25 28 00 mov %fs:0x28,%rax
1c53: 00 00
1c55: 48 89 84 24 08 40 00 mov %rax,0x4008(%rsp)
1c5c: 00
1c5d: 31 c0 xor %eax,%eax
1c5f: 44 8b 05 92 3a 00 00 mov 0x3a92(%rip),%r8d # 56f8
<num_input_strings>
1c66: 41 8d 40 ff lea -0x1(%r8),%eax
1c6a: 48 98 cltq
1c6c: 48 89 c2 mov %rax,%rdx
1c6f: 48 c1 e2 04 shl $0x4,%rdx
1c73: 48 29 c2 sub %rax,%rdx
1c76: 85 ff test %edi,%edi

```



```

1c78:    48 8d 0d b9 17 00 00    lea    0x17b9(%rip),%rcx    # 3438
<array.0+0x258>
1c7f:    48 8d 05 ba 17 00 00    lea    0x17ba(%rip),%rax    # 3440
<array.0+0x260>
1c86:    48 0f 44 c8            cmov    %rax,%rcx
1c8a:    48 89 e3            mov     %rsp,%rbx
1c8d:    48 8d 05 6c 3a 00 00    lea    0x3a6c(%rip),%rax    # 5700
<input_strings>
1c94:    4c 8d 0c d0            lea     (%rax,%rdx,8),%r9
1c98:    8b 15 c6 34 00 00    mov     0x34c6(%rip),%edx    # 5164 <bomb_id>
1c9e:    48 8d 35 a4 17 00 00    lea    0x17a4(%rip),%rsi    # 3449
<array.0+0x269>
1ca5:    48 89 df            mov     %rbx,%rdi
1ca8:    b8 00 00 00 00    mov     $0x0,%eax
1cad:    e8 ce f4 ff ff    call    1180 <sprintf@plt>
1cb2:    4c 8d 84 24 00 20 00    lea    0x2000(%rsp),%r8
1cb9:    00
1cba:    b9 00 00 00 00    mov     $0x0,%ecx
1cbf:    48 89 da            mov     %rbx,%rdx
1cc2:    48 8d 35 77 34 00 00    lea    0x3477(%rip),%rsi    # 5140
<user_password>
1cc9:    48 8d 3d 88 34 00 00    lea    0x3488(%rip),%rdi    # 5158 <userid>
1cd0:    e8 ac 0e 00 00    call    2b81 <driver_post>
1cd5:    85 c0            test    %eax,%eax
1cd7:    78 1c            js      1cf5 <send_msg+0xb1>
1cd9:    48 8b 84 24 08 40 00    mov     0x4008(%rsp),%rax
1ce0:    00
1ce1:    64 48 2b 04 25 28 00    sub     %fs:0x28,%rax
1ce8:    00 00
1cea:    75 20            jne     1d0c <send_msg+0xc8>
1cec:    48 81 c4 10 40 00 00    add     $0x4010,%rsp
1cf3:    5b            pop     %rbx
1cf4:    c3            ret
1cf5:    48 8d bc 24 00 20 00    lea    0x2000(%rsp),%rdi
1cfc:    00
1cfd:    e8 6e f3 ff ff    call    1070 <puts@plt>
1d02:    bf 00 00 00 00    mov     $0x0,%edi
1d07:    e8 84 f4 ff ff    call    1190 <exit@plt>
1d0c:    e8 8f f3 ff ff    call    10a0 <__stack_chk_fail@plt>

```

0000000000001d11 <explode_bomb>:

```
1d11:      48 83 ec 08          sub    $0x8,%rsp
1d15:      48 8d 3d 3c 15 00 00  lea     0x153c(%rip),%rdi      # 3258
<array.0+0x78>
1d1c:      e8 4f f3 ff ff        call   1070 <puts@plt>
1d21:      48 8d 3d 2d 17 00 00  lea     0x172d(%rip),%rdi      # 3455
<array.0+0x275>
1d28:      e8 43 f3 ff ff        call   1070 <puts@plt>
1d2d:      48 8d 3d 3e 17 00 00  lea     0x173e(%rip),%rdi      # 3472
<array.0+0x292>
1d34:      e8 37 f3 ff ff        call   1070 <puts@plt>
1d39:      48 8d 3d 4f 17 00 00  lea     0x174f(%rip),%rdi      # 348f
<array.0+0x2af>
1d40:      e8 2b f3 ff ff        call   1070 <puts@plt>
1d45:      48 8d 3d 60 17 00 00  lea     0x1760(%rip),%rdi      # 34ac
<array.0+0x2cc>
1d4c:      e8 1f f3 ff ff        call   1070 <puts@plt>
1d51:      48 8d 3d 71 17 00 00  lea     0x1771(%rip),%rdi      # 34c9
<array.0+0x2e9>
1d58:      e8 13 f3 ff ff        call   1070 <puts@plt>
1d5d:      48 8d 3d 81 17 00 00  lea     0x1781(%rip),%rdi      # 34e5
<array.0+0x305>
1d64:      e8 07 f3 ff ff        call   1070 <puts@plt>
1d69:      48 8d 3d 92 17 00 00  lea     0x1792(%rip),%rdi      # 3502
<array.0+0x322>
1d70:      e8 fb f2 ff ff        call   1070 <puts@plt>
1d75:      48 8d 3d a3 17 00 00  lea     0x17a3(%rip),%rdi      # 351f
<array.0+0x33f>
1d7c:      e8 ef f2 ff ff        call   1070 <puts@plt>
1d81:      48 8d 3d b4 17 00 00  lea     0x17b4(%rip),%rdi      # 353c
<array.0+0x35c>
1d88:      e8 e3 f2 ff ff        call   1070 <puts@plt>
1d8d:      48 8d 3d c5 17 00 00  lea     0x17c5(%rip),%rdi      # 3559
<array.0+0x379>
1d94:      e8 d7 f2 ff ff        call   1070 <puts@plt>
1d99:      48 8d 3d d6 17 00 00  lea     0x17d6(%rip),%rdi      # 3576
<array.0+0x396>
1da0:      e8 cb f2 ff ff        call   1070 <puts@plt>
1da5:      48 8d 3d e7 17 00 00  lea     0x17e7(%rip),%rdi      # 3593
<array.0+0x3b3>
```

```

1dac:    e8 bf f2 ff ff      call   1070 <puts@plt>
1db1:    bf 00 00 00 00      mov     $0x0,%edi
1db6:    e8 89 fe ff ff      call   1c44 <send_msg>
1dbb:    48 8d 3d de 14 00 00 lea     0x14de(%rip),%rdi    # 32a0
<array.0+0xc0>
1dc2:    e8 a9 f2 ff ff      call   1070 <puts@plt>
1dc7:    bf 08 00 00 00      mov     $0x8,%edi
1dcc:    e8 bf f3 ff ff      call   1190 <exit@plt>

```

0000000000001dd1 <read_six_numbers>:

```

1dd1:    48 83 ec 08          sub     $0x8,%rsp
1dd5:    48 89 f2             mov     %rsi,%rdx //外来传入
1dd8:    48 8d 4e 04          lea     0x4(%rsi),%rcx
1ddc:    48 8d 46 14          lea     0x14(%rsi),%rax
1de0:    50                  push    %rax
1de1:    48 8d 46 10          lea     0x10(%rsi),%rax
1de5:    50                  push    %rax
1de6:    4c 8d 4e 0c          lea     0xc(%rsi),%r9
1dea:    4c 8d 46 08          lea     0x8(%rsi),%r8
1dee:    48 8d 35 b6 17 00 00 lea     0x17b6(%rip),%rsi    # 35ab
<array.0+0x3cb>
1df5:    b8 00 00 00 00      mov     $0x0,%eax
1dfa:    e8 51 f3 ff ff      call   1150 <__isoc99_sscanf@plt>
1dff:    48 83 c4 10          add     $0x10,%rsp
1e03:    83 f8 05             cmp     $0x5,%eax
1e06:    7e 05               jle     1e0d <read_six_numbers+0x3c>
//输入数字<=5 个直接炸
1e08:    48 83 c4 08          add     $0x8,%rsp
1e0c:    c3                  ret
1e0d:    e8 ff fe ff ff      call   1d11 <explode_bomb>

```

0000000000001e12 <read_line>:

```
1e12:    55                push    %rbp
1e13:    53                push    %rbx
1e14:    48 83 ec 08       sub     $0x8,%rsp
1e18:    b8 00 00 00 00    mov     $0x0,%eax
1e1d:    e8 d0 fd ff ff    call    1bf2 <skip>
1e22:    48 85 c0          test    %rax,%rax //skip 返回值是 0 就永远没法跳转
1e25:    74 63            je      1e8a <read_line+0x78>
1e27:    8b 1d cb 38 00 00 mov     0x38cb(%rip),%ebx    # 56f8

<num_input_strings>
1e2d:    48 63 d3          movslq  %ebx,%rdx
1e30:    48 89 d0          mov     %rdx,%rax
1e33:    48 c1 e0 04       shl     $0x4,%rax
1e37:    48 29 d0          sub     %rdx,%rax
1e3a:    48 8d 15 bf 38 00 00 lea     0x38bf(%rip),%rdx    # 5700

<input_strings>
1e41:    48 8d 2c c2       lea     (%rdx,%rax,8),%rbp
1e45:    48 89 ef          mov     %rbp,%rdi
1e48:    e8 43 f2 ff ff    call    1090 <strlen@plt> //第 x 题字符长度 (算上换行)
1e4d:    83 f8 76          cmp     $0x76,%eax //输入长度大于 118 跳转炸弹之路
1e50:    0f 8f ac 00 00 00 jg      1f02 <read_line+0xf0>
1e56:    83 e8 01          sub     $0x1,%eax
1e59:    48 98            cltq
1e5b:    48 63 cb          movslq  %ebx,%rcx
1e5e:    48 89 ca          mov     %rcx,%rdx
1e61:    48 c1 e2 04       shl     $0x4,%rdx //乘以 16
1e65:    48 29 ca          sub     %rcx,%rdx //减 1 倍 ( 也就是乘以 15 )
1e68:    48 8d 0d 91 38 00 00 lea     0x3891(%rip),%rcx    # 5700

<input_strings>
1e6f:    48 8d 14 d1       lea     (%rcx,%rdx,8),%rdx //第 1 题答案 ↑ 第 6 题答案 ↓
1e73:    c6 04 02 00       movb     $0x0,(%rdx,%rax,1) //后者盛放着换行符
1e77:    83 c3 01          add     $0x1,%ebx
1e7a:    89 1d 78 38 00 00 mov     %ebx,0x3878(%rip)    # 56f8

<num_input_strings>
1e80:    48 89 e8          mov     %rbp,%rax //第 x 题答案传入 rax
1e83:    48 83 c4 08       add     $0x8,%rsp
1e87:    5b                pop     %rbx
1e88:    5d                pop     %rbp
1e89:    c3                ret
1e8a:    48 8b 05 bf 37 00 00 mov     0x37bf(%rip),%rax    # 5650
```

<stdin@GLIBC_2.2.5>

```
1e91: 48 39 05 d8 37 00 00    cmp    %rax,0x37d8(%rip)      # 5670 <infile>
1e98: 74 1b                    je     1eb5 <read_line+0xa3>
1e9a: 48 8d 3d 3a 17 00 00    lea     0x173a(%rip),%rdi      # 35db
```

<array.0+0x3fb>

```
1ea1: e8 8a f1 ff ff          call   1030 <getenv@plt>
1ea6: 48 85 c0                test   %rax,%rax
1ea9: 74 20                    je     1ecb <read_line+0xb9>
1eab: bf 00 00 00 00          mov     $0x0,%edi
1eb0: e8 db f2 ff ff          call   1190 <exit@plt>
1eb5: 48 8d 3d 01 17 00 00    lea     0x1701(%rip),%rdi      # 35bd
```

<array.0+0x3dd>

```
1ebc: e8 af f1 ff ff          call   1070 <puts@plt>
1ec1: bf 08 00 00 00          mov     $0x8,%edi
1ec6: e8 c5 f2 ff ff          call   1190 <exit@plt>
1ecb: 48 8b 05 7e 37 00 00    mov     0x377e(%rip),%rax      # 5650
```

<stdin@GLIBC_2.2.5>

//read_line 在 被调用

```
1ed2: 48 89 05 97 37 00 00    mov     %rax,0x3797(%rip)      # 5670 <infile>
1ed9: b8 00 00 00 00          mov     $0x0,%eax
1ede: e8 0f fd ff ff          call   1bf2 <skip>
1ee3: 48 85 c0                test   %rax,%rax
1ee6: 0f 85 3b ff ff ff       jne     1e27 <read_line+0x15>
1eec: 48 8d 3d ca 16 00 00    lea     0x16ca(%rip),%rdi      # 35bd
```

<array.0+0x3dd>

```
1ef3: e8 78 f1 ff ff          call   1070 <puts@plt>
1ef8: bf 00 00 00 00          mov     $0x0,%edi
1efd: e8 8e f2 ff ff          call   1190 <exit@plt>
1f02: 48 8d 3d dd 16 00 00    lea     0x16dd(%rip),%rdi      # 35e6
```

<array.0+0x406>

0x5555555575e6: "Error: Input line too long"

```
1f09: e8 62 f1 ff ff          call   1070 <puts@plt>
1f0e: 8b 05 e4 37 00 00          mov     0x37e4(%rip),%eax      # 56f8
```

<num_input_strings>

```
1f14: 8d 50 01                lea     0x1(%rax),%edx
1f17: 89 15 db 37 00 00          mov     %edx,0x37db(%rip)      # 56f8
```

<num_input_strings>

```
1f1d: 48 98                    cltq
1f1f: 48 6b c0 78             imul    $0x78,%rax,%rax
1f23: 48 8d 15 d6 37 00 00    lea     0x37d6(%rip),%rdx      # 5700
```

<input_strings>

```
1f2a: 48 be 2a 2a 2a 74 72    movabs  $0x636e7572742a2a2a,%rsi
```

```

1f31:    75 6e 63
1f34:    48 bf 61 74 65 64 2a    movabs $0x2a2a2a64657461,%rdi
1f3b:    2a 2a 00
1f3e:    48 89 34 02             mov    %rsi,(%rdx,%rax,1)
1f42:    48 89 7c 02 08         mov    %rdi,0x8(%rdx,%rax,1)
1f47:    e8 c5 fd ff ff         call   1d11 <explode_bomb>

```

0000000000001f4c <phase_defused>:

```

1f4c:    48 83 ec 08             sub    $0x8,%rsp
1f50:    bf 01 00 00 00         mov    $0x1,%edi
1f55:    e8 ea fc ff ff         call   1c44 <send_msg>
1f5a:    83 3d 97 37 00 00 06    cmpl   $0x6,0x3797(%rip) # 56f8 //过第六关才跳转判断

```

<num_input_strings> //必须等于 6 才能不直接退出 这个神秘数字只在 read_line 1ed2 里出现并被赋值

```

1f61:    74 05                  je     1f68 <phase_defused+0x1c>
1f63:    48 83 c4 08           add    $0x8,%rsp
1f67:    c3                    ret
1f68:    0f b6 0d f9 38 00 00    movzbl 0x38f9(%rip),%ecx # 5868

```

<input_strings+0x168>

```

1f6f:    84 c9                 test   %cl,%cl
1f71:    74 34                 je     1fa7 <phase_defused+0x5b>
1f73:    b8 01 00 00 00       mov    $0x1,%eax
1f78:    ba 00 00 00 00       mov    $0x0,%edx
1f7d:    48 8d 3d e4 38 00 00    lea    0x38e4(%rip),%rdi # 5868

```

<input_strings+0x168>

```

1f84:    80 f9 20              cmp     $0x20,%cl
1f87:    0f 94 c1              sete    %cl
1f8a:    0f b6 c9              movzbl %cl,%ecx
1f8d:    01 ca                 add     %ecx,%edx
1f8f:    89 c6                 mov     %eax,%esi
1f91:    0f b6 0c 07           movzbl (%rdi,%rax,1),%ecx
1f95:    48 83 c0 01           add     $0x1,%rax
1f99:    83 fa 01              cmp     $0x1,%edx
1f9c:    7f 04                 jg      1fa2 <phase_defused+0x56> //edx>=1 时跳出
1f9e:    84 c9                 test    %cl,%cl
1fa0:    75 e2                 jne     1f84 <phase_defused+0x39>
1fa2:    83 fa 02              cmp     $0x2,%edx
1fa5:    74 1a                 je      1fc1 <phase_defused+0x75>
1fa7:    48 8d 3d ba 13 00 00    lea     0x13ba(%rip),%rdi # 3368

```

<array.0+0x188>

循环

```

1fae:    e8 bd f0 ff ff      call    1070 <puts@plt>
1fb3:    48 8d 3d de 13 00 00 lea     0x13de(%rip),%rdi      # 3398
<array.0+0x1b8>
1fba:    e8 b1 f0 ff ff      call    1070 <puts@plt>
1fbf:    eb a2               jmp     1f63 <phase_defused+0x17>
1fc1:    48 63 f6            movslq  %esi,%rsi
1fc4:    48 8d 05 9d 38 00 00 lea     0x389d(%rip),%rax      # 5868
<input_strings+0x168>
1fcb:    48 8d 3c 06         lea     (%rsi,%rax,1),%rdi
1fcf:    48 8d 35 f2 12 00 00 lea     0x12f2(%rip),%rsi      # 32c8 <array.0+0xe8>
1fd6:    e8 d1 fa ff ff      call    1aac <strings_not_equal>
1fdb:    85 c0               test    %eax,%eax
1fdd:    75 c8               jne     1fa7 <phase_defused+0x5b>
1fdf:    48 8d 3d 22 13 00 00 lea     0x1322(%rip),%rdi      # 3308
<array.0+0x128>
1fe6:    e8 85 f0 ff ff      call    1070 <puts@plt>
1feb:    48 8d 3d 3e 13 00 00 lea     0x133e(%rip),%rdi      # 3330
<array.0+0x150>
1ff2:    e8 79 f0 ff ff      call    1070 <puts@plt>
1ff7:    b8 00 00 00 00      mov     $0x0,%eax
1ffc:    e8 79 f9 ff ff      call    197a <secret_phase>
2001:    eb a4               jmp     1fa7 <phase_defused+0x5b>

```

0000000000002003 <sigalrm_handler>:

```

2003:    48 83 ec 08         sub     $0x8,%rsp
2007:    ba 00 00 00 00      mov     $0x0,%edx
200c:    48 8d 35 fd 15 00 00 lea     0x15fd(%rip),%rsi      # 3610
<array.0+0x430>
2013:    48 8b 3d 46 36 00 00 mov     0x3646(%rip),%rdi      # 5660
<stderr@GLIBC_2.2.5>

```

```

201a:    b8 00 00 00 00      mov     $0x0,%eax
201f:    e8 0c f1 ff ff      call    1130 <fprintf@plt>
2024:    bf 01 00 00 00      mov     $0x1,%edi
2029:    e8 62 f1 ff ff      call    1190 <exit@plt>

```

000000000000202e <rio_writen>:

```

202e:    41 56               push    %r14
2030:    41 55               push    %r13
2032:    41 54               push    %r12
2034:    55                 push    %rbp

```

2035:	53	push	%rbx
2036:	49 89 d5	mov	%rdx,%r13
2039:	48 85 d2	test	%rdx,%rdx
203c:	74 3b	je	2079 <rio_writen+0x4b>
203e:	41 89 fc	mov	%edi,%r12d
2041:	48 89 f5	mov	%rsi,%rbp
2044:	48 89 d3	mov	%rdx,%rbx
2047:	41 be 00 00 00 00	mov	\$0x0,%r14d
204d:	eb 08	jmp	2057 <rio_writen+0x29>
204f:	48 01 c5	add	%rax,%rbp
2052:	48 29 c3	sub	%rax,%rbx
2055:	74 22	je	2079 <rio_writen+0x4b>
2057:	48 89 da	mov	%rbx,%rdx
205a:	48 89 ee	mov	%rbp,%rsi
205d:	44 89 e7	mov	%r12d,%edi
2060:	e8 1b f0 ff ff	call	1080 <write@plt>
2065:	48 85 c0	test	%rax,%rax
2068:	7f e5	jg	204f <rio_writen+0x21>
206a:	e8 e1 ef ff ff	call	1050 <__errno_location@plt>
206f:	83 38 04	cmpl	\$0x4,(%rax)
2072:	75 11	jne	2085 <rio_writen+0x57>
2074:	4c 89 f0	mov	%r14,%rax
2077:	eb d6	jmp	204f <rio_writen+0x21>
2079:	4c 89 e8	mov	%r13,%rax
207c:	5b	pop	%rbx
207d:	5d	pop	%rbp
207e:	41 5c	pop	%r12
2080:	41 5d	pop	%r13
2082:	41 5e	pop	%r14
2084:	c3	ret	
2085:	48 c7 c0 ff ff ff ff	mov	\$0xffffffffffffffff,%rax
208c:	eb ee	jmp	207c <rio_writen+0x4e>

000000000000208e <rio_readlineb>:

208e:	41 56	push	%r14
2090:	41 55	push	%r13
2092:	41 54	push	%r12
2094:	55	push	%rbp
2095:	53	push	%rbx
2096:	49 89 f4	mov	%rsi,%r12

2099:	48 83 fa 01	cmp	\$0x1,%rdx
209d:	0f 86 92 00 00 00	jbe	2135 <rio_readlineb+0xa7>
20a3:	48 89 fb	mov	%rdi,%rbx
20a6:	4c 8d 74 16 ff	lea	-0x1(%rsi,%rdx,1),%r14
20ab:	41 bd 01 00 00 00	mov	\$0x1,%r13d
20b1:	48 8d 6f 10	lea	0x10(%rdi),%rbp
20b5:	eb 56	jmp	210d <rio_readlineb+0x7f>
20b7:	e8 94 ef ff ff	call	1050 <__errno_location@plt>
20bc:	83 38 04	cmpl	\$0x4,(%rax)
20bf:	75 55	jne	2116 <rio_readlineb+0x88>
20c1:	ba 00 20 00 00	mov	\$0x2000,%edx
20c6:	48 89 ee	mov	%rbp,%rsi
20c9:	8b 3b	mov	(%rbx),%edi
20cb:	e8 10 f0 ff ff	call	10e0 <read@plt>
20d0:	89 c2	mov	%eax,%edx
20d2:	89 43 04	mov	%eax,0x4(%rbx)
20d5:	85 c0	test	%eax,%eax
20d7:	78 de	js	20b7 <rio_readlineb+0x29>
20d9:	85 c0	test	%eax,%eax
20db:	74 42	je	211f <rio_readlineb+0x91>
20dd:	48 89 6b 08	mov	%rbp,0x8(%rbx)
20e1:	48 8b 43 08	mov	0x8(%rbx),%rax
20e5:	0f b6 08	movzbl	(%rax),%ecx
20e8:	48 83 c0 01	add	\$0x1,%rax
20ec:	48 89 43 08	mov	%rax,0x8(%rbx)
20f0:	83 ea 01	sub	\$0x1,%edx
20f3:	89 53 04	mov	%edx,0x4(%rbx)
20f6:	49 83 c4 01	add	\$0x1,%r12
20fa:	41 88 4c 24 ff	mov	%cl,-0x1(%r12)
20ff:	80 f9 0a	cmp	\$0xa,%cl
2102:	74 3c	je	2140 <rio_readlineb+0xb2>
2104:	41 83 c5 01	add	\$0x1,%r13d
2108:	4d 39 f4	cmp	%r14,%r12
210b:	74 30	je	213d <rio_readlineb+0xaf>
210d:	8b 53 04	mov	0x4(%rbx),%edx
2110:	85 d2	test	%edx,%edx
2112:	7e ad	jle	20c1 <rio_readlineb+0x33>
2114:	eb cb	jmp	20e1 <rio_readlineb+0x53>
2116:	48 c7 c0 ff ff ff ff	mov	\$0xffffffffffffffff,%rax
211d:	eb 05	jmp	2124 <rio_readlineb+0x96>

211f:	b8 00 00 00 00	mov	\$0x0,%eax
2124:	85 c0	test	%eax,%eax
2126:	75 29	jne	2151 <rio_readlineb+0xc3>
2128:	b8 00 00 00 00	mov	\$0x0,%eax
212d:	41 83 fd 01	cmp	\$0x1,%r13d
2131:	75 0d	jne	2140 <rio_readlineb+0xb2>
2133:	eb 13	jmp	2148 <rio_readlineb+0xba>
2135:	41 bd 01 00 00 00	mov	\$0x1,%r13d
213b:	eb 03	jmp	2140 <rio_readlineb+0xb2>
213d:	4d 89 f4	mov	%r14,%r12
2140:	41 c6 04 24 00	movb	\$0x0,(%r12)
2145:	49 63 c5	movslq	%r13d,%rax
2148:	5b	pop	%rbx
2149:	5d	pop	%rbp
214a:	41 5c	pop	%r12
214c:	41 5d	pop	%r13
214e:	41 5e	pop	%r14
2150:	c3	ret	
2151:	48 c7 c0 ff ff ff ff	mov	\$0xffffffffffffffff,%rax
2158:	eb ee	jmp	2148 <rio_readlineb+0xba>

000000000000215a <submitr>:

215a:	41 57	push	%r15
215c:	41 56	push	%r14
215e:	41 55	push	%r13
2160:	41 54	push	%r12
2162:	55	push	%rbp
2163:	53	push	%rbx
2164:	48 81 ec 78 a0 00 00	sub	\$0xa078,%rsp
216b:	48 89 7c 24 08	mov	%rdi,0x8(%rsp)
2170:	89 f5	mov	%esi,%ebp
2172:	49 89 d4	mov	%rdx,%r12
2175:	48 89 4c 24 10	mov	%rcx,0x10(%rsp)
217a:	4c 89 44 24 20	mov	%r8,0x20(%rsp)
217f:	4c 89 4c 24 18	mov	%r9,0x18(%rsp)
2184:	48 8b 9c 24 b0 a0 00	mov	0xa0b0(%rsp),%rbx
218b:	00		
218c:	4c 8b bc 24 b8 a0 00	mov	0xa0b8(%rsp),%r15
2193:	00		
2194:	64 48 8b 04 25 28 00	mov	%fs:0x28,%rax

219b:	00 00	
219d:	48 89 84 24 68 a0 00	mov %rax,0xa068(%rsp)
21a4:	00	
21a5:	31 c0	xor %eax,%eax
21a7:	c7 44 24 3c 00 00 00	movl \$0x0,0x3c(%rsp)
21ae:	00	
21af:	ba 00 00 00 00	mov \$0x0,%edx
21b4:	be 01 00 00 00	mov \$0x1,%esi
21b9:	bf 02 00 00 00	mov \$0x2,%edi
21be:	e8 0d f0 ff ff	call 11d0 <socket@plt>
21c3:	85 c0	test %eax,%eax
21c5:	0f 88 0d 01 00 00	js 22d8 <submitr+0x17e>
21cb:	41 89 c6	mov %eax,%r14d
21ce:	48 8b 7c 24 08	mov 0x8(%rsp),%rdi
21d3:	e8 48 ef ff ff	call 1120 <gethostbyname@plt>
21d8:	48 85 c0	test %rax,%rax
21db:	0f 84 47 01 00 00	je 2328 <submitr+0x1ce>
21e1:	4c 8d 6c 24 40	lea 0x40(%rsp),%r13
21e6:	48 c7 44 24 40 00 00	movq \$0x0,0x40(%rsp)
21ed:	00 00	
21ef:	48 c7 44 24 48 00 00	movq \$0x0,0x48(%rsp)
21f6:	00 00	
21f8:	66 c7 44 24 40 02 00	movw \$0x2,0x40(%rsp)
21ff:	48 63 50 14	movslq 0x14(%rax),%rdx
2203:	48 8b 40 18	mov 0x18(%rax),%rax
2207:	48 8d 7c 24 44	lea 0x44(%rsp),%rdi
220c:	48 8b 30	mov (%rax),%rsi
220f:	e8 4c ef ff ff	call 1160 <memmove@plt>
2214:	66 c1 c5 08	rol \$0x8,%bp
2218:	66 89 6c 24 42	mov %bp,0x42(%rsp)
221d:	ba 10 00 00 00	mov \$0x10,%edx
2222:	4c 89 ee	mov %r13,%rsi
2225:	44 89 f7	mov %r14d,%edi
2228:	e8 73 ef ff ff	call 11a0 <connect@plt>
222d:	85 c0	test %eax,%eax
222f:	0f 88 5e 01 00 00	js 2393 <submitr+0x239>
2235:	48 89 df	mov %rbx,%rdi
2238:	e8 53 ee ff ff	call 1090 <strlen@plt>
223d:	48 89 c5	mov %rax,%rbp
2240:	4c 89 e7	mov %r12,%rdi

2243:	e8 48 ee ff ff	call 1090 <strlen@plt>
2248:	49 89 c5	mov %rax,%r13
224b:	48 8b 7c 24 10	mov 0x10(%rsp),%rdi
2250:	e8 3b ee ff ff	call 1090 <strlen@plt>
2255:	49 89 c4	mov %rax,%r12
2258:	48 8b 7c 24 18	mov 0x18(%rsp),%rdi
225d:	e8 2e ee ff ff	call 1090 <strlen@plt>
2262:	48 89 c2	mov %rax,%rdx
2265:	4b 8d 84 25 80 00 00	lea 0x80(%r13,%r12,1),%rax
226c:	00	
226d:	48 01 d0	add %rdx,%rax
2270:	48 8d 54 6d 00	lea 0x0(%rbp,%rbp,2),%rdx
2275:	48 01 d0	add %rdx,%rax
2278:	48 3d 00 20 00 00	cmp \$0x2000,%rax
227e:	0f 87 6c 01 00 00	ja 23f0 <submitr+0x296>
2284:	48 8d 94 24 60 40 00	lea 0x4060(%rsp),%rdx
228b:	00	
228c:	b9 00 04 00 00	mov \$0x400,%ecx
2291:	b8 00 00 00 00	mov \$0x0,%eax
2296:	48 89 d7	mov %rdx,%rdi
2299:	f3 48 ab	rep stos %rax,%es:(%rdi)
229c:	48 89 df	mov %rbx,%rdi
229f:	e8 ec ed ff ff	call 1090 <strlen@plt>
22a4:	85 c0	test %eax,%eax
22a6:	0f 84 13 05 00 00	je 27bf <submitr+0x665>
22ac:	8d 40 ff	lea -0x1(%rax),%eax
22af:	4c 8d 64 03 01	lea 0x1(%rbx,%rax,1),%r12
22b4:	48 8d ac 24 60 40 00	lea 0x4060(%rsp),%rbp
22bb:	00	
22bc:	48 8d 84 24 60 80 00	lea 0x8060(%rsp),%rax
22c3:	00	
22c4:	48 89 44 24 28	mov %rax,0x28(%rsp)
22c9:	49 bd d9 ff 00 00 00	movabs \$0x2000000000ffd9,%r13
22d0:	00 20 00	
22d3:	e9 a5 01 00 00	jmp 247d <submitr+0x323>
22d8:	48 b8 45 72 72 6f 72	movabs \$0x43203a726f727245,%rax
22df:	3a 20 43	
22e2:	48 ba 6c 69 65 6e 74	movabs \$0x6e7520746e65696c,%rdx
22e9:	20 75 6e	
22ec:	49 89 07	mov %rax,(%r15)

22ef:	49 89 57 08	mov	%rdx,0x8(%r15)
22f3:	48 b8 61 62 6c 65 20	movabs	\$0x206f7420656c6261,%rax
22fa:	74 6f 20		
22fd:	48 ba 63 72 65 61 74	movabs	\$0x7320657461657263,%rdx
2304:	65 20 73		
2307:	49 89 47 10	mov	%rax,0x10(%r15)
230b:	49 89 57 18	mov	%rdx,0x18(%r15)
230f:	41 c7 47 20 6f 63 6b	movl	\$0x656b636f,0x20(%r15)
2316:	65		
2317:	66 41 c7 47 24 74 00	movw	\$0x74,0x24(%r15)
231e:	b8 ff ff ff ff	mov	\$0xffffffff,%eax
2323:	e9 6e 03 00 00	jmp	2696 <submitr+0x53c>
2328:	48 b8 45 72 72 6f 72	movabs	\$0x44203a726f727245,%rax
232f:	3a 20 44		
2332:	48 ba 4e 53 20 69 73	movabs	\$0x6e7520736920534e,%rdx
2339:	20 75 6e		
233c:	49 89 07	mov	%rax,(%r15)
233f:	49 89 57 08	mov	%rdx,0x8(%r15)
2343:	48 b8 61 62 6c 65 20	movabs	\$0x206f7420656c6261,%rax
234a:	74 6f 20		
234d:	48 ba 72 65 73 6f 6c	movabs	\$0x2065766c6f736572,%rdx
2354:	76 65 20		
2357:	49 89 47 10	mov	%rax,0x10(%r15)
235b:	49 89 57 18	mov	%rdx,0x18(%r15)
235f:	48 b8 73 65 72 76 65	movabs	\$0x6120726576726573,%rax
2366:	72 20 61		
2369:	49 89 47 20	mov	%rax,0x20(%r15)
236d:	41 c7 47 28 64 64 72	movl	\$0x65726464,0x28(%r15)
2374:	65		
2375:	66 41 c7 47 2c 73 73	movw	\$0x7373,0x2c(%r15)
237c:	41 c6 47 2e 00	movb	\$0x0,0x2e(%r15)
2381:	44 89 f7	mov	%r14d,%edi
2384:	e8 47 ed ff ff	call	10d0 <close@plt>
2389:	b8 ff ff ff ff	mov	\$0xffffffff,%eax
238e:	e9 03 03 00 00	jmp	2696 <submitr+0x53c>
2393:	48 b8 45 72 72 6f 72	movabs	\$0x55203a726f727245,%rax
239a:	3a 20 55		
239d:	48 ba 6e 61 62 6c 65	movabs	\$0x6f7420656c62616e,%rdx
23a4:	20 74 6f		
23a7:	49 89 07	mov	%rax,(%r15)

23aa:	49 89 57 08	mov	%rdx,0x8(%r15)
23ae:	48 b8 20 63 6f 6e 6e	movabs	\$0x7463656e6e6f6320,%rax
23b5:	65 63 74		
23b8:	48 ba 20 74 6f 20 74	movabs	\$0x20656874206f7420,%rdx
23bf:	68 65 20		
23c2:	49 89 47 10	mov	%rax,0x10(%r15)
23c6:	49 89 57 18	mov	%rdx,0x18(%r15)
23ca:	41 c7 47 20 73 65 72	movl	\$0x76726573,0x20(%r15)
23d1:	76		
23d2:	66 41 c7 47 24 65 72	movw	\$0x7265,0x24(%r15)
23d9:	41 c6 47 26 00	movb	\$0x0,0x26(%r15)
23de:	44 89 f7	mov	%r14d,%edi
23e1:	e8 ea ec ff ff	call	10d0 <close@plt>
23e6:	b8 ff ff ff ff	mov	\$0xffffffff,%eax
23eb:	e9 a6 02 00 00	jmp	2696 <submitr+0x53c>
23f0:	48 b8 45 72 72 6f 72	movabs	\$0x52203a726f727245,%rax
23f7:	3a 20 52		
23fa:	48 ba 65 73 75 6c 74	movabs	\$0x747320746c757365,%rdx
2401:	20 73 74		
2404:	49 89 07	mov	%rax,(%r15)
2407:	49 89 57 08	mov	%rdx,0x8(%r15)
240b:	48 b8 72 69 6e 67 20	movabs	\$0x6f6f7420676e6972,%rax
2412:	74 6f 6f		
2415:	48 ba 20 6c 61 72 67	movabs	\$0x202e656772616c20,%rdx
241c:	65 2e 20		
241f:	49 89 47 10	mov	%rax,0x10(%r15)
2423:	49 89 57 18	mov	%rdx,0x18(%r15)
2427:	48 b8 49 6e 63 72 65	movabs	\$0x6573616572636e49,%rax
242e:	61 73 65		
2431:	48 ba 20 53 55 42 4d	movabs	\$0x5254494d42555320,%rdx
2438:	49 54 52		
243b:	49 89 47 20	mov	%rax,0x20(%r15)
243f:	49 89 57 28	mov	%rdx,0x28(%r15)
2443:	48 b8 5f 4d 41 58 42	movabs	\$0x46554258414d5f,%rax
244a:	55 46 00		
244d:	49 89 47 30	mov	%rax,0x30(%r15)
2451:	44 89 f7	mov	%r14d,%edi
2454:	e8 77 ec ff ff	call	10d0 <close@plt>
2459:	b8 ff ff ff ff	mov	\$0xffffffff,%eax
245e:	e9 33 02 00 00	jmp	2696 <submitr+0x53c>

2463:	49 0f a3 c5	bt	%rax,%r13	
2467:	73 1e	jae	2487 <submitr+0x32d>	
2469:	88 55 00	mov	%dl,0x0(%rbp)	
246c:	48 8d 6d 01	lea	0x1(%rbp),%rbp	
2470:	48 83 c3 01	add	\$0x1,%rbx	
2474:	4c 39 e3	cmp	%r12,%rbx	
2477:	0f 84 42 03 00 00	je	27bf <submitr+0x665>	
247d:	0f b6 13	movzbl	(%rbx),%edx	
2480:	8d 42 d6	lea	-0x2a(%rdx),%eax	
2483:	3c 35	cmp	\$0x35,%al	
2485:	76 dc	jbe	2463 <submitr+0x309>	
2487:	89 d0	mov	%edx,%eax	
2489:	83 e0 df	and	\$0xffffffff,%eax	
248c:	83 e8 41	sub	\$0x41,%eax	
248f:	3c 19	cmp	\$0x19,%al	
2491:	76 d6	jbe	2469 <submitr+0x30f>	
2493:	80 fa 20	cmp	\$0x20,%dl	
2496:	74 50	je	24e8 <submitr+0x38e>	
2498:	8d 42 e0	lea	-0x20(%rdx),%eax	
249b:	3c 5f	cmp	\$0x5f,%al	
249d:	76 09	jbe	24a8 <submitr+0x34e>	
249f:	80 fa 09	cmp	\$0x9,%dl	
24a2:	0f 85 8a 02 00 00	jne	2732 <submitr+0x5d8>	
24a8:	0f b6 d2	movzbl	%dl,%edx	
24ab:	48 8d 35 35 12 00 00	lea	0x1235(%rip),%rsi	# 36e7
<array.0+0x507>				
24b2:	48 8b 7c 24 28	mov	0x28(%rsp),%rdi	
24b7:	b8 00 00 00 00	mov	\$0x0,%eax	
24bc:	e8 bf ec ff ff	call	1180 <sprintf@plt>	
24c1:	0f b6 84 24 60 80 00	movzbl	0x8060(%rsp),%eax	
24c8:	00			
24c9:	88 45 00	mov	%al,0x0(%rbp)	
24cc:	0f b6 84 24 61 80 00	movzbl	0x8061(%rsp),%eax	
24d3:	00			
24d4:	88 45 01	mov	%al,0x1(%rbp)	
24d7:	0f b6 84 24 62 80 00	movzbl	0x8062(%rsp),%eax	
24de:	00			
24df:	88 45 02	mov	%al,0x2(%rbp)	
24e2:	48 8d 6d 03	lea	0x3(%rbp),%rbp	
24e6:	eb 88	jmp	2470 <submitr+0x316>	

24e8:	c6 45 00 2b	movb \$0x2b,0x0(%rbp)
24ec:	48 8d 6d 01	lea 0x1(%rbp),%rbp
24f0:	e9 7b ff ff ff	jmp 2470 <submitr+0x316>
24f5:	48 b8 45 72 72 6f 72	movabs \$0x43203a726f727245,%rax
24fc:	3a 20 43	
24ff:	48 ba 6c 69 65 6e 74	movabs \$0x6e7520746e65696c,%rdx
2506:	20 75 6e	
2509:	49 89 07	mov %rax,(%r15)
250c:	49 89 57 08	mov %rdx,0x8(%r15)
2510:	48 b8 61 62 6c 65 20	movabs \$0x206f7420656c6261,%rax
2517:	74 6f 20	
251a:	48 ba 77 72 69 74 65	movabs \$0x6f74206574697277,%rdx
2521:	20 74 6f	
2524:	49 89 47 10	mov %rax,0x10(%r15)
2528:	49 89 57 18	mov %rdx,0x18(%r15)
252c:	48 b8 20 74 68 65 20	movabs \$0x7265732065687420,%rax
2533:	73 65 72	
2536:	49 89 47 20	mov %rax,0x20(%r15)
253a:	41 c7 47 28 76 65 72	movl \$0x726576,0x28(%r15)
2541:	00	
2542:	44 89 f7	mov %r14d,%edi
2545:	e8 86 eb ff ff	call 10d0 <close@plt>
254a:	b8 ff ff ff ff	mov \$0xffffffff,%eax
254f:	e9 42 01 00 00	jmp 2696 <submitr+0x53c>
2554:	48 b8 45 72 72 6f 72	movabs \$0x43203a726f727245,%rax
255b:	3a 20 43	
255e:	48 ba 6c 69 65 6e 74	movabs \$0x6e7520746e65696c,%rdx
2565:	20 75 6e	
2568:	49 89 07	mov %rax,(%r15)
256b:	49 89 57 08	mov %rdx,0x8(%r15)
256f:	48 b8 61 62 6c 65 20	movabs \$0x206f7420656c6261,%rax
2576:	74 6f 20	
2579:	48 ba 77 72 69 74 65	movabs \$0x6f74206574697277,%rdx
2580:	20 74 6f	
2583:	49 89 47 10	mov %rax,0x10(%r15)
2587:	49 89 57 18	mov %rdx,0x18(%r15)
258b:	48 b8 20 74 68 65 20	movabs \$0x7265732065687420,%rax
2592:	73 65 72	
2595:	49 89 47 20	mov %rax,0x20(%r15)
2599:	41 c7 47 28 76 65 72	movl \$0x726576,0x28(%r15)


```

25a0:    00
25a1:    44 89 f7             mov     %r14d,%edi
25a4:    e8 27 eb ff ff      call   10d0 <close@plt>
25a9:    b8 ff ff ff ff      mov     $0xffffffff,%eax
25ae:    e9 e3 00 00 00      jmp     2696 <submitr+0x53c>
25b3:    48 b8 45 72 72 6f 72 movabs  $0x43203a726f727245,%rax
25ba:    3a 20 43
25bd:    48 ba 6c 69 65 6e 74 movabs  $0x6e7520746e65696c,%rdx
25c4:    20 75 6e
25c7:    49 89 07             mov     %rax,(%r15)
25ca:    49 89 57 08          mov     %rdx,0x8(%r15)
25ce:    48 b8 61 62 6c 65 20 movabs  $0x206f7420656c6261,%rax
25d5:    74 6f 20
25d8:    48 ba 72 65 61 64 20 movabs  $0x7269662064616572,%rdx
25df:    66 69 72
25e2:    49 89 47 10          mov     %rax,0x10(%r15)
25e6:    49 89 57 18          mov     %rdx,0x18(%r15)
25ea:    48 b8 73 74 20 68 65 movabs  $0x6564616568207473,%rax
25f1:    61 64 65
25f4:    48 ba 72 20 66 72 6f movabs  $0x73206d6f72662072,%rdx
25fb:    6d 20 73
25fe:    49 89 47 20          mov     %rax,0x20(%r15)
2602:    49 89 57 28          mov     %rdx,0x28(%r15)
2606:    41 c7 47 30 65 72 76 movl    $0x65767265,0x30(%r15)
260d:    65
260e:    66 41 c7 47 34 72 00 movw    $0x72,0x34(%r15)
2615:    44 89 f7             mov     %r14d,%edi
2618:    e8 b3 ea ff ff      call   10d0 <close@plt>
261d:    b8 ff ff ff ff      mov     $0xffffffff,%eax
2622:    eb 72               jmp     2696 <submitr+0x53c>
2624:    48 8d 8c 24 60 80 00 lea     0x8060(%rsp),%rcx
262b:    00
262c:    48 8d 35 05 10 00 00 lea     0x1005(%rip),%rsi      # 3638
<array.0+0x458>
2633:    4c 89 ff             mov     %r15,%rdi
2636:    b8 00 00 00 00      mov     $0x0,%eax
263b:    e8 40 eb ff ff      call   1180 <sprintf@plt>
2640:    44 89 f7             mov     %r14d,%edi
2643:    e8 88 ea ff ff      call   10d0 <close@plt>
2648:    b8 ff ff ff ff      mov     $0xffffffff,%eax

```

```

264d:    eb 47                jmp     2696 <submitr+0x53c>
264f:    48 8d b4 24 60 20 00 lea     0x2060(%rsp),%rsi
2656:    00
2657:    48 8d 7c 24 50        lea     0x50(%rsp),%rdi
265c:    ba 00 20 00 00        mov     $0x2000,%edx
2661:    e8 28 fa ff ff        call    208e <rio_readlineb>
2666:    48 85 c0              test    %rax,%rax
2669:    7e 54                jle     26bf <submitr+0x565>
266b:    48 8d b4 24 60 20 00 lea     0x2060(%rsp),%rsi
2672:    00
2673:    4c 89 ff             mov     %r15,%rdi
2676:    e8 e5 e9 ff ff        call    1060 <strcpy@plt>
267b:    44 89 f7             mov     %r14d,%edi
267e:    e8 4d ea ff ff        call    10d0 <close@plt>
2683:    48 8d 35 82 10 00 00 lea     0x1082(%rip),%rsi      # 370c
<array.0+0x52c>
268a:    4c 89 ff             mov     %r15,%rdi
268d:    e8 6e ea ff ff        call    1100 <strcmp@plt>
2692:    f7 d8              neg     %eax
2694:    19 c0              sbb     %eax,%eax
2696:    48 8b 94 24 68 a0 00 mov     0xa068(%rsp),%rdx
269d:    00
269e:    64 48 2b 14 25 28 00 sub     %fs:0x28,%rdx
26a5:    00 00
26a7:    0f 85 be 02 00 00     jne     296b <submitr+0x811>
26ad:    48 81 c4 78 a0 00 00 add     $0xa078,%rsp
26b4:    5b                pop     %rbx
26b5:    5d                pop     %rbp
26b6:    41 5c              pop     %r12
26b8:    41 5d              pop     %r13
26ba:    41 5e              pop     %r14
26bc:    41 5f              pop     %r15
26be:    c3                ret
26bf:    48 b8 45 72 72 6f 72 movabs  $0x43203a726f727245,%rax
26c6:    3a 20 43
26c9:    48 ba 6c 69 65 6e 74 movabs  $0x6e7520746e65696c,%rdx
26d0:    20 75 6e
26d3:    49 89 07             mov     %rax,(%r15)
26d6:    49 89 57 08          mov     %rdx,0x8(%r15)
26da:    48 b8 61 62 6c 65 20 movabs  $0x206f7420656c6261,%rax

```

26e1:	74 6f 20	
26e4:	48 ba 72 65 61 64 20	movabs \$0x6174732064616572,%rdx
26eb:	73 74 61	
26ee:	49 89 47 10	mov %rax,0x10(%r15)
26f2:	49 89 57 18	mov %rdx,0x18(%r15)
26f6:	48 b8 74 75 73 20 6d	movabs \$0x7373656d20737574,%rax
26fd:	65 73 73	
2700:	48 ba 61 67 65 20 66	movabs \$0x6d6f726620656761,%rdx
2707:	72 6f 6d	
270a:	49 89 47 20	mov %rax,0x20(%r15)
270e:	49 89 57 28	mov %rdx,0x28(%r15)
2712:	48 b8 20 73 65 72 76	movabs \$0x72657672657320,%rax
2719:	65 72 00	
271c:	49 89 47 30	mov %rax,0x30(%r15)
2720:	44 89 f7	mov %r14d,%edi
2723:	e8 a8 e9 ff ff	call 10d0 <close@plt>
2728:	b8 ff ff ff ff	mov \$0xffffffff,%eax
272d:	e9 64 ff ff ff	jmp 2696 <submitr+0x53c>
2732:	48 b8 45 72 72 6f 72	movabs \$0x52203a726f727245,%rax
2739:	3a 20 52	
273c:	48 ba 65 73 75 6c 74	movabs \$0x747320746c757365,%rdx
2743:	20 73 74	
2746:	49 89 07	mov %rax,(%r15)
2749:	49 89 57 08	mov %rdx,0x8(%r15)
274d:	48 b8 72 69 6e 67 20	movabs \$0x6e6f6320676e6972,%rax
2754:	63 6f 6e	
2757:	48 ba 74 61 69 6e 73	movabs \$0x6e6120736e696174,%rdx
275e:	20 61 6e	
2761:	49 89 47 10	mov %rax,0x10(%r15)
2765:	49 89 57 18	mov %rdx,0x18(%r15)
2769:	48 b8 20 69 6c 6c 65	movabs \$0x6c6167656c6c6920,%rax
2770:	67 61 6c	
2773:	48 ba 20 6f 72 20 75	movabs \$0x72706e7520726f20,%rdx
277a:	6e 70 72	
277d:	49 89 47 20	mov %rax,0x20(%r15)
2781:	49 89 57 28	mov %rdx,0x28(%r15)
2785:	48 b8 69 6e 74 61 62	movabs \$0x20656c6261746e69,%rax
278c:	6c 65 20	
278f:	48 ba 63 68 61 72 61	movabs \$0x6574636172616863,%rdx
2796:	63 74 65	

```

2799:    49 89 47 30          mov     %rax,0x30(%r15)
279d:    49 89 57 38          mov     %rdx,0x38(%r15)
27a1:    66 41 c7 47 40 72 2e  movw    $0x2e72,0x40(%r15)
27a8:    41 c6 47 42 00        movb    $0x0,0x42(%r15)
27ad:    44 89 f7             mov     %r14d,%edi
27b0:    e8 1b e9 ff ff        call    10d0 <close@plt>
27b5:    b8 ff ff ff ff        mov     $0xffffffff,%eax
27ba:    e9 d7 fe ff ff        jmp     2696 <submitr+0x53c>
27bf:    48 8d 9c 24 60 20 00   lea     0x2060(%rsp),%rbx
27c6:    00
27c7:    4c 8d 8c 24 60 40 00   lea     0x4060(%rsp),%r9
27ce:    00
27cf:    4c 8b 44 24 18        mov     0x18(%rsp),%r8
27d4:    48 8b 4c 24 20        mov     0x20(%rsp),%rcx
27d9:    48 8b 54 24 10        mov     0x10(%rsp),%rdx
27de:    48 8d 35 83 0e 00 00   lea     0xe83(%rip),%rsi      # 3668
<array.0+0x488>
27e5:    48 89 df             mov     %rbx,%rdi
27e8:    b8 00 00 00 00        mov     $0x0,%eax
27ed:    e8 8e e9 ff ff        call    1180 <sprintf@plt>
27f2:    48 89 df             mov     %rbx,%rdi
27f5:    e8 96 e8 ff ff        call    1090 <strlen@plt>
27fa:    48 89 c2             mov     %rax,%rdx
27fd:    48 89 de             mov     %rbx,%rsi
2800:    44 89 f7             mov     %r14d,%edi
2803:    e8 26 f8 ff ff        call    202e <rio_writen>
2808:    48 85 c0             test    %rax,%rax
280b:    0f 88 e4 fc ff ff     js      24f5 <submitr+0x39b>
2811:    48 8d 9c 24 60 20 00   lea     0x2060(%rsp),%rbx
2818:    00
2819:    48 8b 54 24 08        mov     0x8(%rsp),%rdx
281e:    48 8d 35 c9 0e 00 00   lea     0xec9(%rip),%rsi      # 36ee
<array.0+0x50e>
2825:    48 89 df             mov     %rbx,%rdi
2828:    b8 00 00 00 00        mov     $0x0,%eax
282d:    e8 4e e9 ff ff        call    1180 <sprintf@plt>
2832:    48 89 df             mov     %rbx,%rdi
2835:    e8 56 e8 ff ff        call    1090 <strlen@plt>
283a:    48 89 c2             mov     %rax,%rdx
283d:    48 89 de             mov     %rbx,%rsi

```

```

2840:    44 89 f7                mov     %r14d,%edi
2843:    e8 e6 f7 ff ff         call    202e <rio_writen>
2848:    48 85 c0                test    %rax,%rax
284b:    0f 88 03 fd ff ff       js      2554 <submitr+0x3fa>
2851:    44 89 74 24 50          mov     %r14d,0x50(%rsp)
2856:    c7 44 24 54 00 00 00    movl    $0x0,0x54(%rsp)
285d:    00
285e:    48 8d 7c 24 50          lea     0x50(%rsp),%rdi
2863:    48 8d 44 24 60          lea     0x60(%rsp),%rax
2868:    48 89 44 24 58          mov     %rax,0x58(%rsp)
286d:    48 8d b4 24 60 20 00    lea     0x2060(%rsp),%rsi
2874:    00
2875:    ba 00 20 00 00          mov     $0x2000,%edx
287a:    e8 0f f8 ff ff         call    208e <rio_readlineb>
287f:    48 85 c0                test    %rax,%rax
2882:    0f 8e 2b fd ff ff       jle     25b3 <submitr+0x459>
2888:    48 8d 4c 24 3c          lea     0x3c(%rsp),%rcx
288d:    48 8d 94 24 60 60 00    lea     0x6060(%rsp),%rdx
2894:    00
2895:    48 8d bc 24 60 20 00    lea     0x2060(%rsp),%rdi
289c:    00
289d:    4c 8d 84 24 60 80 00    lea     0x8060(%rsp),%r8
28a4:    00
28a5:    48 8d 35 4f 0e 00 00    lea     0xe4f(%rip),%rsi      # 36fb
<array.0+0x51b>
28ac:    b8 00 00 00 00          mov     $0x0,%eax
28b1:    e8 9a e8 ff ff         call    1150 <__isoc99_sscanf@plt>
28b6:    8b 54 24 3c            mov     0x3c(%rsp),%edx
28ba:    81 fa c8 00 00 00       cmp     $0xc8,%edx
28c0:    0f 85 5e fd ff ff       jne     2624 <submitr+0x4ca>
28c6:    48 8d 1d 2b 0e 00 00    lea     0xe2b(%rip),%rbx      # 36f8
<array.0+0x518>
28cd:    48 8d bc 24 60 20 00    lea     0x2060(%rsp),%rdi
28d4:    00
28d5:    48 89 de                mov     %rbx,%rsi
28d8:    e8 23 e8 ff ff         call    1100 <strcmp@plt>
28dd:    85 c0                  test    %eax,%eax
28df:    0f 84 6a fd ff ff       je      264f <submitr+0x4f5>
28e5:    48 8d b4 24 60 20 00    lea     0x2060(%rsp),%rsi
28ec:    00

```

```

28ed:    48 8d 7c 24 50        lea    0x50(%rsp),%rdi
28f2:    ba 00 20 00 00        mov    $0x2000,%edx
28f7:    e8 92 f7 ff ff        call   208e <rio_readlineb>
28fc:    48 85 c0               test   %rax,%rax
28ff:    7f cc                 jg     28cd <submitr+0x773>
2901:    48 b8 45 72 72 6f 72   movabs $0x43203a726f727245,%rax
2908:    3a 20 43
290b:    48 ba 6c 69 65 6e 74   movabs $0x6e7520746e65696c,%rdx
2912:    20 75 6e
2915:    49 89 07               mov    %rax,(%r15)
2918:    49 89 57 08            mov    %rdx,0x8(%r15)
291c:    48 b8 61 62 6c 65 20   movabs $0x206f7420656c6261,%rax
2923:    74 6f 20
2926:    48 ba 72 65 61 64 20   movabs $0x6165682064616572,%rdx
292d:    68 65 61
2930:    49 89 47 10            mov    %rax,0x10(%r15)
2934:    49 89 57 18            mov    %rdx,0x18(%r15)
2938:    48 b8 64 65 72 73 20   movabs $0x6f72662073726564,%rax
293f:    66 72 6f
2942:    48 ba 6d 20 73 65 72   movabs $0x726576726573206d,%rdx
2949:    76 65 72
294c:    49 89 47 20            mov    %rax,0x20(%r15)
2950:    49 89 57 28            mov    %rdx,0x28(%r15)
2954:    41 c6 47 30 00         movb   $0x0,0x30(%r15)
2959:    44 89 f7               mov    %r14d,%edi
295c:    e8 6f e7 ff ff        call   10d0 <close@plt>
2961:    b8 ff ff ff ff        mov    $0xffffffff,%eax
2966:    e9 2b fd ff ff        jmp    2696 <submitr+0x53c>
296b:    e8 30 e7 ff ff        call   10a0 <__stack_chk_fail@plt>

```

0000000000002970 <init_timeout>:

```

2970:    85 ff                 test   %edi,%edi
2972:    75 01                 jne    2975 <init_timeout+0x5>
2974:    c3                   ret
2975:    53                   push   %rbx
2976:    89 fb                 mov    %edi,%ebx
2978:    48 8d 35 84 f6 ff ff   lea    -0x97c(%rip),%rsi    # 2003

```

<sigalrm_handler>

```

297f:    bf 0e 00 00 00        mov    $0xe,%edi
2984:    e8 87 e7 ff ff        call   1110 <signal@plt>

```

```

2989:      85 db          test   %ebx,%ebx
298b:      b8 00 00 00 00 mov    $0x0,%eax
2990:      0f 49 c3       cmovns %ebx,%eax
2993:      89 c7         mov    %eax,%edi
2995:      e8 26 e7 ff ff call   10c0 <alarm@plt>
299a:      5b            pop    %rbx
299b:      c3           ret

```

000000000000299c <init_driver>:

```

299c:      41 54          push   %r12
299e:      55            push   %rbp
299f:      53            push   %rbx
29a0:      48 83 ec 20     sub    $0x20,%rsp
29a4:      48 89 fd       mov    %rdi,%rbp
29a7:      64 48 8b 04 25 28 00 mov    %fs:0x28,%rax
29ae:      00 00
29b0:      48 89 44 24 18     mov    %rax,0x18(%rsp)
29b5:      31 c0          xor    %eax,%eax
29b7:      be 01 00 00 00     mov    $0x1,%esi
29bc:      bf 0d 00 00 00     mov    $0xd,%edi
29c1:      e8 4a e7 ff ff     call   1110 <signal@plt>
29c6:      be 01 00 00 00     mov    $0x1,%esi
29cb:      bf 1d 00 00 00     mov    $0x1d,%edi
29d0:      e8 3b e7 ff ff     call   1110 <signal@plt>
29d5:      be 01 00 00 00     mov    $0x1,%esi
29da:      bf 1d 00 00 00     mov    $0x1d,%edi
29df:      e8 2c e7 ff ff     call   1110 <signal@plt>
29e4:      ba 00 00 00 00     mov    $0x0,%edx
29e9:      be 01 00 00 00     mov    $0x1,%esi
29ee:      bf 02 00 00 00     mov    $0x2,%edi
29f3:      e8 d8 e7 ff ff     call   11d0 <socket@plt>
29f8:      85 c0          test   %eax,%eax
29fa:      0f 88 97 00 00 00 js     2a97 <init_driver+0xfb>
2a00:      89 c3         mov    %eax,%ebx
2a02:      48 8d 3d 06 0d 00 00 lea     0xd06(%rip),%rdi      # 370f

```

<array.0+0x52f>

```

2a09:      e8 12 e7 ff ff     call   1120 <gethostbyname@plt>
2a0e:      48 85 c0          test   %rax,%rax
2a11:      0f 84 cc 00 00 00 je     2ae3 <init_driver+0x147>
2a17:      49 89 e4         mov    %rsp,%r12

```

2a1a:	48 c7 04 24 00 00 00	movq \$0x0,(%rsp)
2a21:	00	
2a22:	48 c7 44 24 08 00 00	movq \$0x0,0x8(%rsp)
2a29:	00 00	
2a2b:	66 c7 04 24 02 00	movw \$0x2,(%rsp)
2a31:	48 63 50 14	movslq 0x14(%rax),%rdx
2a35:	48 8b 40 18	mov 0x18(%rax),%rax
2a39:	48 8d 7c 24 04	lea 0x4(%rsp),%rdi
2a3e:	48 8b 30	mov (%rax),%rsi
2a41:	e8 1a e7 ff ff	call 1160 <memmove@plt>
2a46:	66 c7 44 24 02 00 50	movw \$0x5000,0x2(%rsp)
2a4d:	ba 10 00 00 00	mov \$0x10,%edx
2a52:	4c 89 e6	mov %r12,%rsi
2a55:	89 df	mov %ebx,%edi
2a57:	e8 44 e7 ff ff	call 11a0 <connect@plt>
2a5c:	85 c0	test %eax,%eax
2a5e:	0f 88 e7 00 00 00	js 2b4b <init_driver+0x1af>
2a64:	89 df	mov %ebx,%edi
2a66:	e8 65 e6 ff ff	call 10d0 <close@plt>
2a6b:	66 c7 45 00 4f 4b	movw \$0x4b4f,0x0(%rbp)
2a71:	c6 45 02 00	movb \$0x0,0x2(%rbp)
2a75:	b8 00 00 00 00	mov \$0x0,%eax
2a7a:	48 8b 54 24 18	mov 0x18(%rsp),%rdx
2a7f:	64 48 2b 14 25 28 00	sub %fs:0x28,%rdx
2a86:	00 00	
2a88:	0f 85 ee 00 00 00	jne 2b7c <init_driver+0x1e0>
2a8e:	48 83 c4 20	add \$0x20,%rsp
2a92:	5b	pop %rbx
2a93:	5d	pop %rbp
2a94:	41 5c	pop %r12
2a96:	c3	ret
2a97:	48 b8 45 72 72 6f 72	movabs \$0x43203a726f727245,%rax
2a9e:	3a 20 43	
2aa1:	48 ba 6c 69 65 6e 74	movabs \$0x6e7520746e65696c,%rdx
2aa8:	20 75 6e	
2aab:	48 89 45 00	mov %rax,0x0(%rbp)
2aaf:	48 89 55 08	mov %rdx,0x8(%rbp)
2ab3:	48 b8 61 62 6c 65 20	movabs \$0x206f7420656c6261,%rax
2aba:	74 6f 20	
2abd:	48 ba 63 72 65 61 74	movabs \$0x7320657461657263,%rdx


```

2ac4:    65 20 73
2ac7:    48 89 45 10          mov     %rax,0x10(%rbp)
2acb:    48 89 55 18          mov     %rdx,0x18(%rbp)
2acf:    c7 45 20 6f 63 6b 65  movl    $0x656b636f,0x20(%rbp)
2ad6:    66 c7 45 24 74 00     movw    $0x74,0x24(%rbp)
2adc:    b8 ff ff ff          mov     $0xffffffff,%eax
2ae1:    eb 97                jmp     2a7a <init_driver+0xde>
2ae3:    48 b8 45 72 72 6f 72  movabs  $0x44203a726f727245,%rax
2aea:    3a 20 44
2aed:    48 ba 4e 53 20 69 73  movabs  $0x6e7520736920534e,%rdx
2af4:    20 75 6e
2af7:    48 89 45 00          mov     %rax,0x0(%rbp)
2afb:    48 89 55 08          mov     %rdx,0x8(%rbp)
2aff:    48 b8 61 62 6c 65 20  movabs  $0x206f7420656c6261,%rax
2b06:    74 6f 20
2b09:    48 ba 72 65 73 6f 6c  movabs  $0x2065766c6f736572,%rdx
2b10:    76 65 20
2b13:    48 89 45 10          mov     %rax,0x10(%rbp)
2b17:    48 89 55 18          mov     %rdx,0x18(%rbp)
2b1b:    48 b8 73 65 72 76 65  movabs  $0x6120726576726573,%rax
2b22:    72 20 61
2b25:    48 89 45 20          mov     %rax,0x20(%rbp)
2b29:    c7 45 28 64 64 72 65  movl    $0x65726464,0x28(%rbp)
2b30:    66 c7 45 2c 73 73     movw    $0x7373,0x2c(%rbp)
2b36:    c6 45 2e 00          movb    $0x0,0x2e(%rbp)
2b3a:    89 df                mov     %ebx,%edi
2b3c:    e8 8f e5 ff ff       call    10d0 <close@plt>
2b41:    b8 ff ff ff ff       mov     $0xffffffff,%eax
2b46:    e9 2f ff ff ff       jmp     2a7a <init_driver+0xde>
2b4b:    b9 50 00 00 00       mov     $0x50,%ecx
2b50:    48 8d 15 b8 0b 00 00  lea     0xbb8(%rip),%rdx      # 370f
<array.0+0x52f>
2b57:    48 8d 35 5a 0b 00 00  lea     0xb5a(%rip),%rsi      # 36b8
<array.0+0x4d8>
2b5e:    48 89 ef             mov     %rbp,%rdi
2b61:    b8 00 00 00 00       mov     $0x0,%eax
2b66:    e8 15 e6 ff ff       call    1180 <sprintf@plt>
2b6b:    89 df                mov     %ebx,%edi
2b6d:    e8 5e e5 ff ff       call    10d0 <close@plt>
2b72:    b8 ff ff ff ff       mov     $0xffffffff,%eax

```

```

2b77:    e9 fe fe ff ff      jmp     2a7a <init_driver+0xde>
2b7c:    e8 1f e5 ff ff      call    10a0 <__stack_chk_fail@plt>

```

0000000000002b81 <driver_post>:

```

2b81:    53                  push    %rbx
2b82:    4c 89 c3            mov     %r8,%rbx
2b85:    85 c9              test    %ecx,%ecx
2b87:    75 17              jne     2ba0 <driver_post+0x1f>
2b89:    48 85 ff            test    %rdi,%rdi
2b8c:    74 05              je      2b93 <driver_post+0x12>
2b8e:    80 3f 00            cmpb    $0x0,(%rdi)
2b91:    75 31              jne     2bc4 <driver_post+0x43>
2b93:    66 c7 03 4f 4b      movw    $0x4b4f,(%rbx)
2b98:    c6 43 02 00         movb    $0x0,0x2(%rbx)
2b9c:    89 c8              mov     %ecx,%eax
2b9e:    5b                pop     %rbx
2b9f:    c3                ret
2ba0:    48 89 d6            mov     %rdx,%rsi
2ba3:    48 8d 3d 70 0b 00 00 lea     0xb70(%rip),%rdi      # 371a

```

<array.0+0x53a>

```

2baa:    b8 00 00 00 00      mov     $0x0,%eax
2baf:    e8 fc e4 ff ff      call    10b0 <printf@plt>
2bb4:    66 c7 03 4f 4b      movw    $0x4b4f,(%rbx)
2bb9:    c6 43 02 00         movb    $0x0,0x2(%rbx)
2bbd:    b8 00 00 00 00      mov     $0x0,%eax
2bc2:    eb da              jmp     2b9e <driver_post+0x1d>
2bc4:    41 50              push    %r8
2bc6:    52                push    %rdx
2bc7:    4c 8d 0d 63 0b 00 00 lea     0xb63(%rip),%r9      # 3731

```

<array.0+0x551>

```

2bce:    49 89 f0            mov     %rsi,%r8
2bd1:    48 89 f9            mov     %rdi,%rcx
2bd4:    48 8d 15 5e 0b 00 00 lea     0xb5e(%rip),%rdx      # 3739

```

<array.0+0x559>

```

2bdb:    be 50 00 00 00      mov     $0x50,%esi
2be0:    48 8d 3d 28 0b 00 00 lea     0xb28(%rip),%rdi      # 370f

```

<array.0+0x52f>

```

2be7:    e8 6e f5 ff ff      call    215a <submitr>
2bec:    48 83 c4 10          add     $0x10,%rsp
2bf0:    eb ac              jmp     2b9e <driver_post+0x1d>

```

```

2bf2:    66 2e 0f 1f 84 00 00    cs nopw 0x0(%rax,%rax,1)
2bf9:    00 00 00
2bfc:    0f 1f 40 00            nopl    0x0(%rax)

```

0000000000002c00 <__libc_csu_init>:

```

2c00:    f3 0f 1e fa            endbr64
2c04:    41 57                  push    %r15
2c06:    4c 8d 3d db 21 00 00    lea     0x21db(%rip),%r15    # 4de8

```

<__frame_dummy_init_array_entry>

```

2c0d:    41 56                  push    %r14
2c0f:    49 89 d6              mov     %rdx,%r14
2c12:    41 55                  push    %r13
2c14:    49 89 f5              mov     %rsi,%r13
2c17:    41 54                  push    %r12
2c19:    41 89 fc              mov     %edi,%r12d
2c1c:    55                    push    %rbp
2c1d:    48 8d 2d cc 21 00 00    lea     0x21cc(%rip),%rbp    # 4df0

```

<__do_global_ctors_aux_fini_array_entry>

```

2c24:    53                    push    %rbx
2c25:    4c 29 fd              sub     %r15,%rbp
2c28:    48 83 ec 08           sub     $0x8,%rsp
2c2c:    e8 cf e3 ff ff        call    1000 <_init>
2c31:    48 c1 fd 03           sar     $0x3,%rbp
2c35:    74 1f                 je      2c56 <__libc_csu_init+0x56>
2c37:    31 db                 xor     %ebx,%ebx
2c39:    0f 1f 80 00 00 00 00    nopl    0x0(%rax)
2c40:    4c 89 f2              mov     %r14,%rdx
2c43:    4c 89 ee              mov     %r13,%rsi
2c46:    44 89 e7              mov     %r12d,%edi
2c49:    41 ff 14 df           call    *(%r15,%rbx,8)
2c4d:    48 83 c3 01           add     $0x1,%rbx
2c51:    48 39 dd              cmp     %rbx,%rbp
2c54:    75 ea                 jne     2c40 <__libc_csu_init+0x40>
2c56:    48 83 c4 08           add     $0x8,%rsp
2c5a:    5b                    pop     %rbx
2c5b:    5d                    pop     %rbp
2c5c:    41 5c                 pop     %r12
2c5e:    41 5d                 pop     %r13
2c60:    41 5e                 pop     %r14
2c62:    41 5f                 pop     %r15

```

2c64:	c3	ret
2c65:	66 66 2e 0f 1f 84 00	data16 cs nopw 0x0(%rax,%rax,1)
2c6c:	00 00 00 00	

0000000000002c70 <__libc_csu_fini>:

2c70:	f3 0f 1e fa	endbr64
2c74:	c3	ret

Disassembly of section .fini:

0000000000002c78 <_fini>:

2c78:	f3 0f 1e fa	endbr64
2c7c:	48 83 ec 08	sub \$0x8,%rsp
2c80:	48 83 c4 08	add \$0x8,%rsp
2c84:	c3	ret