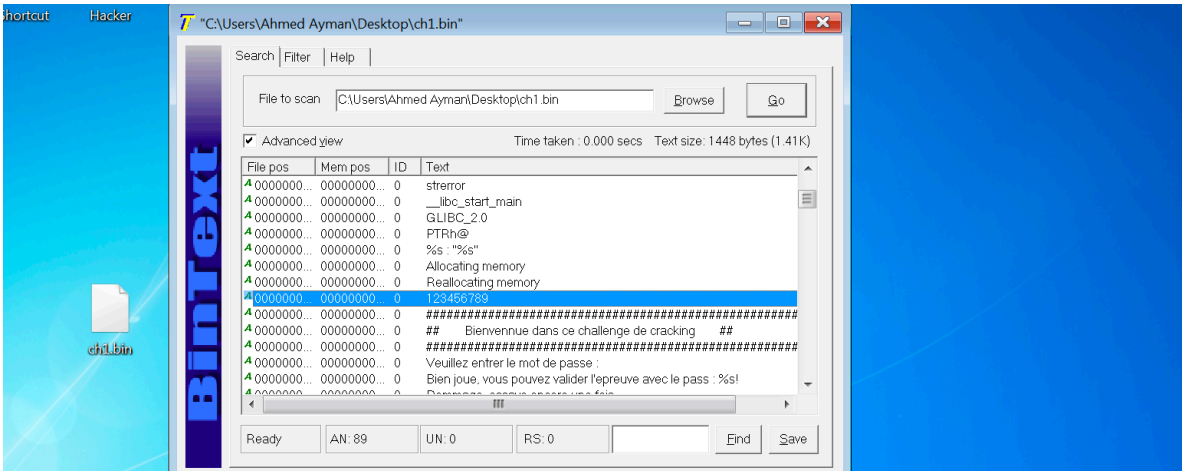
Reverse

ELF x86 - 0 protection

First challenge of cracking, writen in C with vi and compiled with GCC32



using bintext.txt or strings in linux

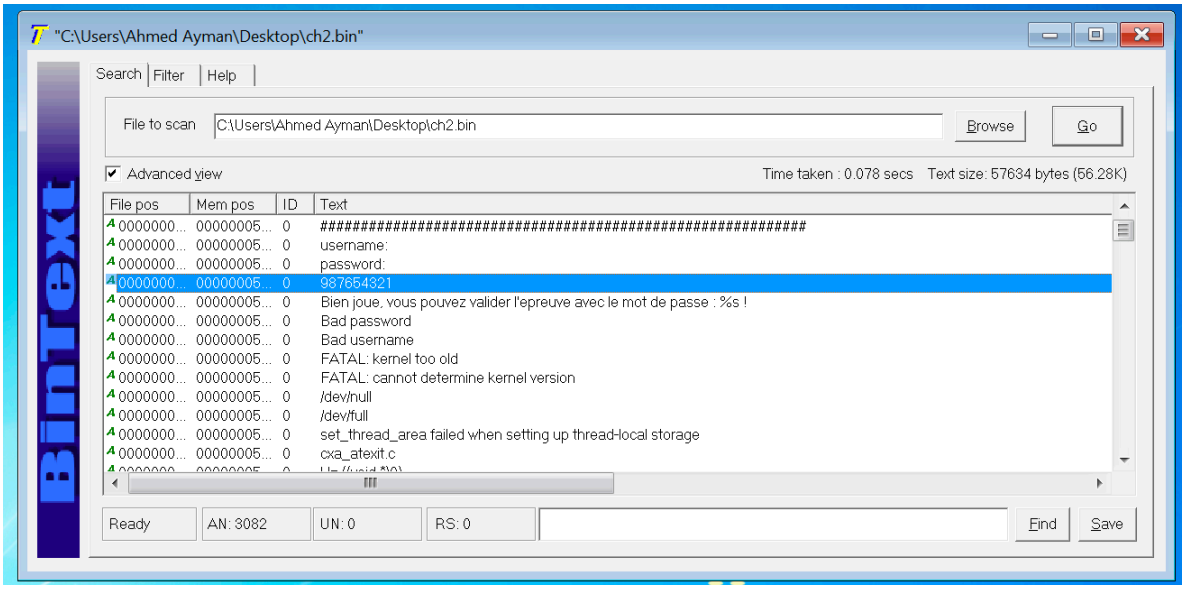
flag = 123456789

other solutions

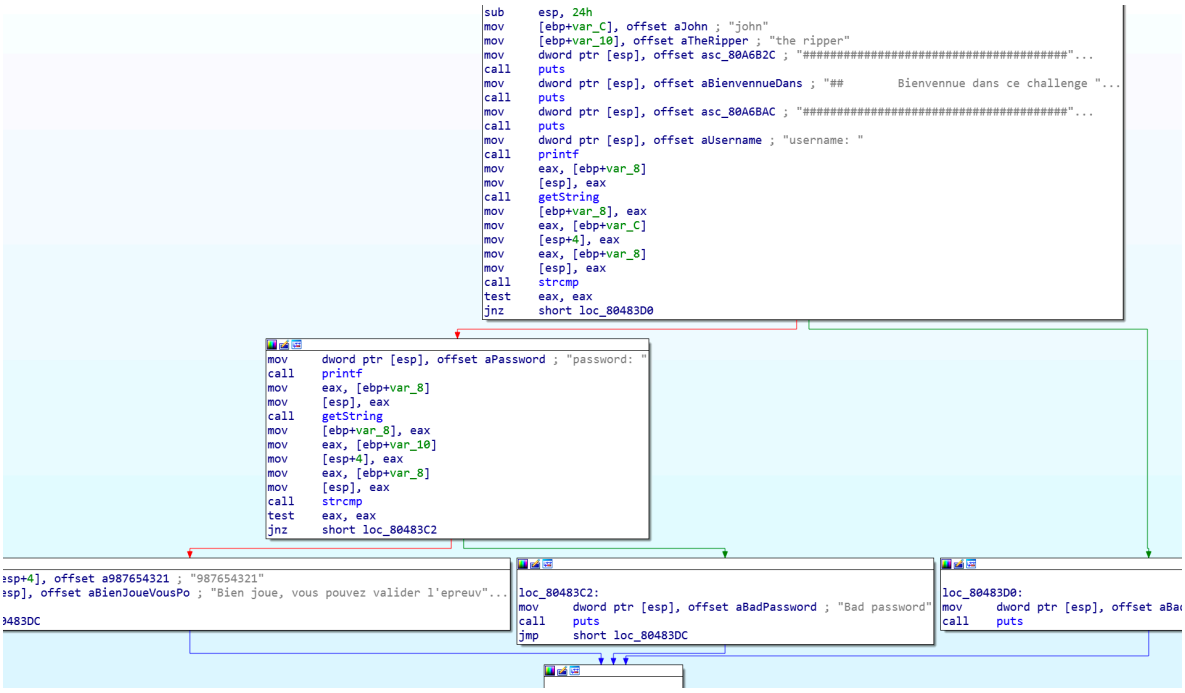
1. $ **file** ch1.bin
2. ch1.bin: ELF 32-bit LSB executable, Intel 80386, version 1 **(**SYSV**)**, dynamically linked, interpreter **/**lib**/**ld-linux.so.2, **for** GNU**/**Linux 2.6.9, not stripped

rabin2 -I ch1.bin
rabin2 -z ch1.bin

ELF x86 - Basic



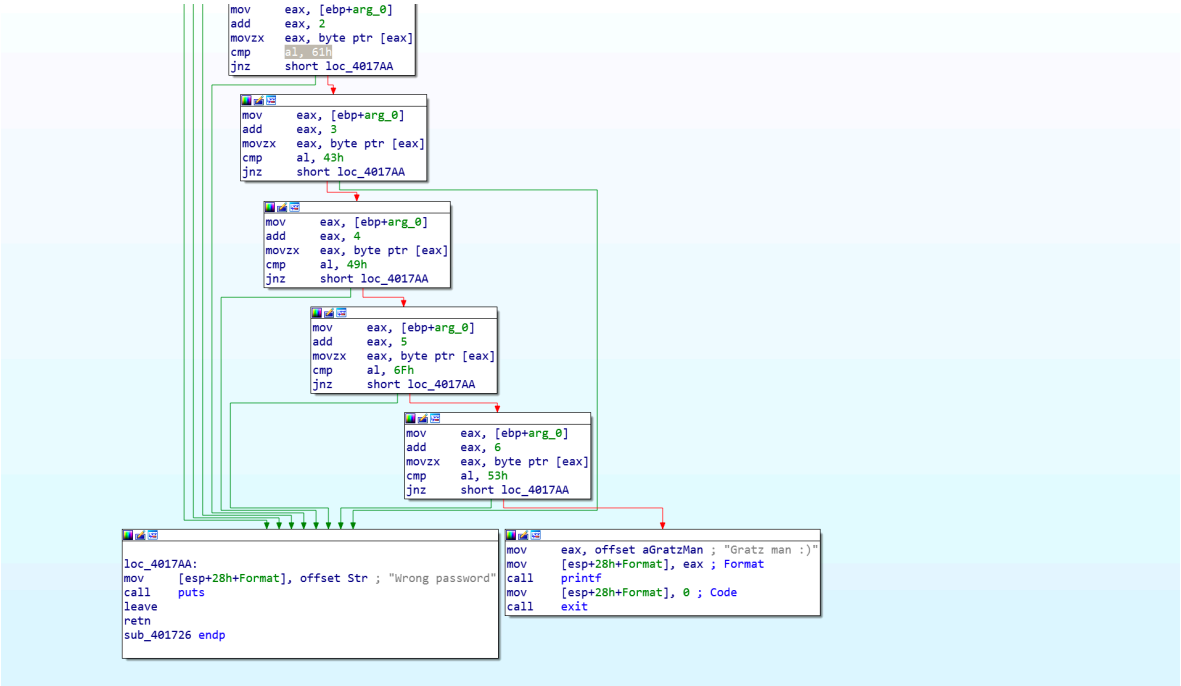
with IDA-free



PE x86 - 0 protection

- Open the file in IDA
- Search for wrong password string and you'll see sub_401726
- In this function, first it checks length of string with 7, if it's not 7 it shows wrong password error, so it should be 7 chars.
- Step by step it checks password with a hex code, as you see in picture, it starts with 0x53, then 0x50, then 0x61.... and so on. Build this string which is the password.

SPaCIoS



ELF C++ - 0 protection

decompile with ghidra

navigation --> go to --> main

```
local_10 = &param_1;
if (param_1 < 2) {
  pcVar1 = *param_2;
  this = operator<<<std--char_traits<char>>((basic_ostream *)cerr,"usage : ");
  this = operator<<<std--char_traits<char>>(this,pcVar1);
  this = operator<<<std--char_traits<char>>(this," password");
  operator<<((basic_ostream<char,std--char_traits<char>> *)this,
             _ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_);
  uVar3 = 5;
}
else {
  allocator();
                    /* try { // try from 08048b0e to 08048b12 has its CatchHandler @ 08048c64 */
  basic_string((char *)local_14,(allocator *)&DAT_08048dc4);
  allocator();
                    /* try { // try from 08048b33 to 08048b37 has its CatchHandler @ 08048c3b */
  basic_string((char *)local_18,(allocator *)&DAT_08048dcc);
                    /* try { // try from 08048b4c to 08048b50 has its CatchHandler @ 08048c1d */
  plouf((basic_string)&local_1c,(basic_string)local_18);
                    /* try { // try from 08048b5a to 08048b5e has its CatchHandler @ 08048c2c */
  ~basic_string(local_18);
  ~allocator(&local_1e);
                    /* try { // try from 08048b70 to 08048b74 has its CatchHandler @ 08048c55 */
  ~basic_string(local_14);
  ~allocator(&local_1d);
                    /* try { // try from 08048b92 to 08048c08 has its CatchHandler @ 08048c7b */
  bVar2 = operator===<char,std--char_traits<char>,std--allocator<char>>(&local_1c,param_2[1]);
  if (bVar2 == false) {
    this = operator<<<std--char_traits<char>>((basic_ostream *)cout,"Password incorrect.");
    operator<<((basic_ostream<char,std--char_traits<char>> *)this,
               _ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_);
```

navigation --> go to. --> plouf

https://re.kv.io/crackme/4.html