**Homework Assignment #4**

Q> 다음에 제시된 코드는 모두 정수 취약점이 존재한다. ① 어떠한 취약점인지 적고 이유를 설명하세요 ② 정상적으로 동작할 경우와 취약점으로 인해 비정상적으로 동작할 경우를 각각 실행하고 실행화면을 제시하세요. 비정상적인 경우만 있을 때에는 그 경우만 제시하세요. ③ 발견된 취약점을 완화할 수 있는 전략을 설명하고 코드를 고치세요 ④ ②에서 비정상적으로 동작한 경우에 대해 실행하고 정상적으로 동작하는 모습을 캡처화면으로 제시하세요.

Question 1) --------------------------------------------------------------------------------------

#include <stdio.h>

#include <string.h>

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

unsigned short s;

int i;

char buf[80];

if(argc < 3){

return -1;

}

i = atoi(argv[1]);

s = i;

if(s >= 80){ /\* [w1] \*/

printf("Too big memory allocation!\n");

return -1;

}

printf("s = %d\n", s);

memcpy(buf, argv[2], i);

buf[i] = '\0';

printf("%s\n", buf);

return 0;

}

Question 2) --------------------------------------------------------------------------------------

|  |
| --- |
| #include <stdio.h> |
| #include <limits.h> |
| #include <assert.h>  int main(void) {  assert(sizeof(unsigned char)==1);  unsigned char uc1 = 0xff;  unsigned char uc2 = 0;  if(~uc1 == uc2) {  printf("%08x == %08x\n", ~uc1, uc2);  } else {  printf("%08x != %08x\n", ~uc1, uc2);  }  return 0;  } |
|  |

Question 3) --------------------------------------------------------------------------------------

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

int main(int argc, char \*\*argv)

{

int val, i;

char \*mem;

if (argc < 2)

exit(1);

val = atoi(argv[1]);

if (val > 0) {

mem = malloc(val \* sizeof(char \*));

if (mem == NULL) {

printf("Failure\n");

exit(2);

}

}

for (i = 0; i < val; i++) {

mem[i] = 'A';

printf("%c", mem[i]);

}

printf("\n");

return 0;

}