[SW보안 개론- 3분반] 6월 24일, 두 번째 단기 진행과제

다음 물음에 번호 순으로 답하시오. (총 12문제 110점) 풀이과정이나 원리 설명이 더 중요합니다.

10:30~11:40분까지 보고서 정리하고, 11:45분까지 PDF로 생성하여 email로 제출해야 합니다.

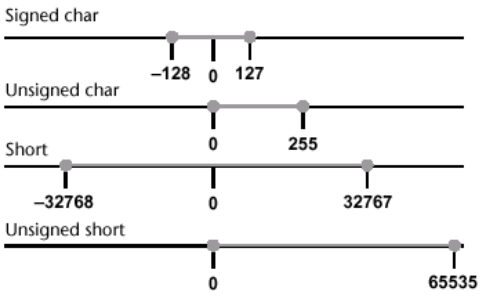
1. Software bug와 software vulnerability의 차이점은? (5점)

2. char 와 short 자료형들의 범위가 오른쪽과 같을 때, 다음 프로그램들의 출력 결과를 보이고, 왜 그러한 결과가 나오는지에 대해 상세히 설명하시오. 출력 결과 자체 보다는 왜 그러한 결과가 나오는지를 설명하는 것이 훨씬 더 중요 합니다. (10점)

```
#include <stdio.h>    // my_char_2.c

void main()
{
        unsigned short us1 = 65535;
        short s1 = 0x7FFF;
        char c = -1;
        int r1, r2;

        r1 = r2 = 0;
        if(c == us1)
                r1 = printf("Why is -1 == 65535 ???\n");
        if(c < us1)
                r2 = printf("Why is -1 < 65535 ???\n");
        printf("0x%x, 0x%x\n", c, us1);
        printf("%d, %d\n", r1, r2);
        printf("s1 + 1 = %d, %hd, 0x%x\n", s1+1, s1+1, s1+1);
}
```



3. 다음 프로그램의 출력 결과를 보이고, 왜 그러한 결과가 나오는지에 대해 상세히 설명하시오. (15점)

```
#include <stdio.h>      // my_short_3.c

void main()
{
        int j = 0xbeefcafe;
        unsigned short k = 0xFFFF;
        short m = k;
        short n = 0x800;
        char c;

        printf("%d(0x%x), %d(0x%x)\n", k, k, m, m);
        printf("%d(0x%x), %d(0x%x)\n", k+1, k+1, m+1, m+1);
        printf("%hd(0hx%hx), %hd(0hx%hx)\n", k+1, k+1, m+1, m+1);
        printf("%hd(0x%x), %hd(0x%x)\n", k-1, k-1, m-1, m-1);

        m = 2*n;
        printf("2 * n = %hd(0x%x) \n", m, m);

        m = j; c = j;
        printf("m = %d(0x%x), c = %d(0x%x) \n", m, m, c, c);
}
```

앞 5개의 printf()문에서는 10진수와 16진수 출력을 정확하게 보이시오.

단, 마지막 printf()문에서 m의 10진수 값은 양수인지 음수인지 설명만 해도 됩니다. (16진수 출력은 정확하게 보이고, 16진수로 출력되는 숫자가 양수인지 음수인지 밝히시오.)

4. 다음 프로그램의 출력 결과는? 그 이유도 같이 설명하시오. (5점)

```
#include <stdio.h>
void main()
{       int k = 1;              int val = 0;

        while (k = 10) {
                k++;            val++;
                if(val > 20) break;
        }
        printf("k = %d, val = %d\n", k, val);
}
```

5. stack_guard.c를 아래와 작성한 다음, 인텔 64-bit 구조의 Ubuntu에서 컴파일하여, stack_guard0를 생성하였다. 버퍼 오버플로우 버그를 악용하여 '**you win!**'을 출력할 수 있게 하는 gets()의 입력을 보여야 합니다. python을 사용한 입력 값과 파이프라인(|)을 사용하시오. (10점)

$ gcc -fno-stack-protector -o stack_guard0 stack_guard.c

```
#include <stdio.h>      /* stack_guard.c */
#include <string.h>
#define goodPass "GOODPASS"
int main () {
        char passIsGood = 0;
        short canary = 35;
        char canary2 = '3';
        char buf[32];
        printf("%08x, %x, %x, %08x\n", buf, &canary2, &canary, &passIsGood);
        printf("Enter password: \n");
        gets(buf);
        if(canary != 35 || canary2 != '3'){
                printf("BOF attack!\n");
                return(-1);
        }
        if(strcmp (buf, goodPass)==0) passIsGood =5;
        if(passIsGood == 5)
                printf("you win!\n");
        return 0;
}
```

참고:
지역변수의 주소는 다음과 같다고 가정한다.

&passIsGood: 0x9c096eef
&canary: 0x9c096eec
&canary2: 0x9c096eeb
Buf: 0x9c096ec0

입력의 총 길이는 몇 바이트이어야 하는지도 설명하시오.

6. 다음 물음에 답하시오. (8점)

우측 프로그램("my_cp.c") 프로그램이 왜 보안에 취약한지 설명하시오.

$ make my_cp

위와 같이 컴파일 한 후에, 실제 실행 예를 보이시오. 즉, 아래와 같이 실행할 때, OOO 와 ???에 어떤 값들이 주어질 때 위험한지 구체적인 예를 보이시오.

$ ./my_cp OOO ???

```
1.  #include <stdio.h> /* my_cp.c */
2.  #include <stdlib.h>
3.
4.  int main(int argc, char *argv[]) {
5.      int retval;
6.      char buffer[50];
7.
8.      if (argc < 3) {
9.          printf("usage: my_cp src_file dst_file\n");
10.         return -1;
11.     }
12.     sprintf(buffer, "cp %s %s", argv[1], argv[2]);
13.     system(buffer);
14. }
```

7. 다음 프로그램의 실행 결과를 보이시오. (17점)

```
#include <stdio.h>    // arrays_3.c
void main() {
        int *ptr;
        int a[5] = {0, 1, 2, 3, 4};
        int b[6] = {5, 6, 7, 8, 9, 10};
        int c[7] = {11, 12, 13, 14, 15, 16, 17};
        void print_arr(int *, unsigned short);

        printf("a= 0x%x, b=0x%x, c=0x%x\n", a, b, c);
        ptr = b+1;
        ptr[-6] = 100;          ptr[-4] = 200;
        ptr[-2] = 300;          ptr[0] = 400;
        ptr[2] = 500;           ptr[4] = 600;
        ptr[6] = 700;           ptr[8] = 800;
        ptr[10] = 900;
        print_arr(a, 5);
        print_arr(b, 6);
        print_arr(c, 7);
}
void print_arr(int *arr, unsigned short k) {
        for(int i=0; i<k; i++)
                printf("%4d ", arr[i]);
        printf("\n");
}
```

```
$ gcc –o arrays_3 arrays_3.c
$ ./arrays_3

a[]의 시작 주소: 0xe9cea5e0
b[]의 시작주소: 0xe9cea600
c[]의 시작주소: 0xe9cea620
a= 0xe9cea5e0, b=0xe9cea600, c=0xe9cea620
ptr에는 b+1 가 할당됩니다.

출력을 보이고, 왜 그렇게 출력되는지 그 이유도 함께 설명하시오.
```

8. Secure SDLC에서 요구사항 분석에는 security requirements를 포함한다. Good password에 대한 security requirements를 3개 이상 설명하시오. (6점)

9. Microsoft SDL에서 Threat modeling이란 무엇인지 자세히 설명하시오. (10점)

10. 컴퓨터 바이러스와 컴퓨터 웜의 차이점을 표를 그려 설명하시오. (6점)

11. Ransomware와 Cryptojacking malware에 대해 설명하고, 차이점을 자세하게 설명하시오. (이전 homework에서 나온 문제임) (8점)

12. Malware analysis에서 Static analysis와 Dynamic analysis의 차이점, 그리고 장단점에 대해 자세히 설명하시오. (10점)