[SW보안 개론- 2분반] 6월 24일, 두 번째 단기 진행과제

다음 물음에 번호 순으로 답하시오. (총 11문제 110점) 풀이과정이나 원리 설명이 더 중요합니다.

13:00~14:10분까지 보고서 정리하고, 14:15분까지 PDF로 생성하여 email로 제출해야 합니다.

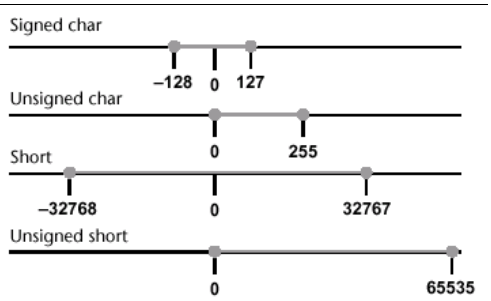
1. char, short 자료형의 범위가 오른쪽과 같을 때, 다음 프로그램들의 출력 결과를 보이고, 왜 그러한 결과가 나오는지에 대해 상세히 설명하시오. 출력 결과 자체 보다는 왜 그러한 결과가 나오는지를 설명하는 것이 훨씬 더 중요 합니다. (12점)

```
#include <stdio.h>      // my_char_3.c

void main()
{
        char c1, c2, csub, csum;

        c1 = 100; c2 = 128;
        csub = c1-c2;      csum = c1+c2;

        printf("%hd(0hx%hx), %hd(0hx%hx) \n", c1, c1, c2, c2);
        printf("%hhd(0hhx%hx, %hhd(0hhx%hhx) \n", c1, c1, c2, c2);
        printf("%hd(0hx%hx), %hd(0hx%hx) \n", csub, csub, csum, csum);
        printf("%hhd(0hhx%hhx), %hd(0hhx%hhx) \n", csub, csub, csum,
csum);
}
```



2. 다음 프로그램의 출력 결과를 보이고, 왜 그러한 결과가 나오는지에 대해 상세히 설명하시오. (13점)

```
include <stdio.h>    // my_short_2.c

void main()
{
        int j = 0xcafe6789;
        unsigned short k = 0x7FFF;
        short m = k;
        short n = 0x8000;
        char c;

        printf("%d(0x%x), %d(0x%x)\n", k, k, m, m);
        printf("%d(0x%x), %d(0x%x)\n", k+2, k+2, m+2, m+2);
        printf("%hu(0hx%hx), %hd(0hx%hx)\n", k+2, k+2, m+2, m+2);

        m = n - 0xFFFF;
        printf("n - 0xFFFF = %hd(0hx%hx) \n", m, m);

        m = j; c = j;
        printf("m = %hd(0hx%hx), c = %hhd(0hhx%hhx) \n", m, m, c, c);
}
```

시간이 부족하면 10진수는 양수인지 음수인지 설명만 해도 됩니다. 단, 16진수 출력은 정확하게 보이고, 16진수에 대응되는 10진수가 양수인지 음수인지 밝히시오.

왜 그렇게 출력되는지를 함께 설명하시오.

3. 다음 프로그램 수행 후, 출력되는 값은? (5점)

```
#include <stdio.h>    // my_arr_2.c
void main()
{
        unsigned short k;
        char array[5] = {'a', 'a', 'a', 'a', 'a'};

        for(k=1; k<=5; k++)
                array[k] = 'a'+k;
        array[k] = 'B';
        for(int i=0; i<=k; i++)
                printf("%2c ", array[i]);
        printf("\n");
}
```

```
$ gcc –o my_arr_2 my_arr_2.c
$ ./my_arr_2

수행 결과는 →
```

4. 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 = 40;
        int    canary2 = 0xff;
        char buf[38];

        printf("%08x, %x, %x, %08x\n", buf, &canary2, &canary, &passIsGood);
        printf("Enter password: \n");
        gets(buf);

        printf("canary2 = 0x%x, canary= 0x%x\n", canary2, canary);

        if(canary != 40 || canary2 != 0xff){
                printf("BOF attack!\n");
                return(-1);
        }
        if(strcmp (buf, goodPass)==0) passIsGood = '3';
        if(passIsGood == '3')
                printf("you win!\n");
        return 0;
}
```

참고:
주소는 다음과 같다.

&passIsGood: 0xc96af1ff
&canary: 0xc96af1fc
&canary2: 0xc96af1f8
Buf: 0xc96af1d0

입력의 총 길이는 몇 바이트이어야 하는지도 설명하시오.

5. 삼성전자나 Naver, Google 등이 software bug bounty program을 운영하는 이유는? (5점)

6. Microsoft SDL에서 Static analysis와 Dynamic analysis의 차이점, 그리고 장단점에 자세히 설명하시오. Fuzz testing에 대해서도 설명하시오. (10점)

7. SW보안에서 penetrate and patch 접근방법이란? 이 접근방법의 문제점에 대해 자세히 설명하시오? 또한 이 문제를 해결하기 위한 방안도 설명하시오. (10점)

8. 다음 프로그램의 실행 결과를 보이시오. (15점)

```
#include <stdio.h>    // arrays_3.c
void main() {
        short *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 = (short *)b;
        ptr[-12] = 100;          ptr[-8] = 200;
        ptr[-4] = 300;           ptr[0] = 400;
        ptr[4] = 500;            ptr[8] = 600;
        ptr[12] = 700;           ptr[16] = 800;
        ptr[18] = 900;           ptr[20] = 1000;
        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: 0x5d23b120
b[]의 시작주소: 0x5d23b140
c[]의 시작주소: 0x5d23b160
a= 0x5d23b120, b=0x5d23b140, c=0x5d23b160

ptr의 타입은 "short *" 입니다.
출력을 보이고, 왜 그렇게 출력되는지 그 이유도 함께 설명하시오.
```

9. OWASP Top 10 Vulnerabilities 중에서 Injection에 대해 설명하시오. Injection과 관련된 CWE를 4개 이상 나열하여 함께 설명하시오. (10점)

10. 컴퓨터 웜과 Zombies (Bots)을 특징을 설명하고 비교하시오. (5점)

11. Malware 은닉 기법을 5가지 나열하고, 각각 자세하게 설명하시오. (15점)