



20225261

bao.kv225261@sis.hust.edu.vn

1

2

3

4

5

6

7

8

9

Quiz 14 - Lỗi hỏng truy cập bộ nhớ (2)

Homework due Jul 1, 2025 20:00 +07 *Completed*

Question #18529

1/1 point (graded)

Kích hoạt chế độ non-executable stack không thể phòng chống được các hành vi tấn công tràn bộ đệm nào sau đây?

- ☐ Thực thi mã nguồn trong vùng nhớ stack
- ☒ Thực thi hàm thư viện chuẩn cung cấp lời gọi hệ thống
- ☒ Thực thi các đoạn mã nguồn đã có sẵn trong bộ nhớ
- ☒ Ghi đè dữ liệu vào vùng nhớ khác
- ☒ Thực thi mã nguồn trong vùng nhớ heap



Submit

Question #bfd18

1/1 point (graded)

Cho hàm viết bằng chương trình C như sau. Khai báo biến canh giữ đặt ở dòng nào để phòng chống được các hành vi tấn công khai thác lỗi hỏng tràn bộ đệm?

```
1
2 int n;
3
4 float x;
5
6 int array1[64];
7
8 float array2[32];
9
```

- ☐ 1
- ☐ 3
- ☒ 5
- ☒ 7
- ☐ 9



Submit

Question #1a668

1/1 point (graded)

Cho hàm viết bằng chương trình C như sau. Lỗi hỏng tràn bộ đệm có thể sửa như thế nào?

```
void func(char buff1[], char buff2[])
{
    int n = strlen(buff2);
    if(n < 1024)
        strcpy(buff1, buff2);
}
```

- ☐ Xóa dòng lệnh số 3
- ☐ Xóa dòng lệnh số 2
- ☒ Sửa điều kiện dòng lệnh số 4 thành if (strlen(buff1) > strlen(buff2))
- ☐ Sửa điều kiện dòng lệnh số 4 thành if (n <= 1024)



Submit

#### Question #2758b

1/1 point (graded)

Giả sử khai báo biến canh giữ canary được khai báo là kiểu số nguyên 4 byte. Kẻ tấn công phải thử trung bình bao nhiêu trường hợp để có thể tấn công thành công?

- ☐ 256
- ☐ 4
- ☐ 32
- ☒  $2^{31}$
- ☐  $2^{32}$



Submit

#### Question #e2819

1/1 point (graded)

Tính năng ASLR(Address Space Layout Randomization) trên các hệ điều hành là gì?

- ☒ Sử dụng địa chỉ ngẫu nhiên cho các vùng nhớ trong bộ nhớ ảo của tiến trình
- ☐ Cất các thông tin của lời gọi hàm vào vị trí ngẫu nhiên trong stack
- ☐ Lưu trữ các file hệ thống ở vị trí ngẫu nhiên trên ổ cứng mỗi khi khởi động
- ☐ Tất cả các đáp án



Submit

#### Question #43aa6

1/1 point (graded)

Câu 6. Cho đoạn chương trình viết bằng C sau đây. Lỗi hỏng xâu định dạng có thể được sửa như thế nào?

```
1 char name[64];
2 gets(name);
3 printf(name);
```

- ☐ Xóa dòng lệnh số 3
- ☐ Sửa dòng lệnh số 3 thành if(strlen(name) > 0) printf(name);

- ☐ Xóa dòng lệnh số 2
- ☐ Sửa dòng lệnh số 2 thành `fgets(name, 1024, stdin);`
- ☒ Sửa dòng lệnh số 3 thành `printf("%s", name);`



Submit

#### Question #7c047

1/1 point (graded)

Câu 7. Trường hợp nào sau đây không truy cập bộ nhớ an toàn về không gian?

- ☒ Ghi đè dữ liệu sang vùng nhớ ngoài vùng nhớ chứa biến
- ☐ Truy cập vào các ô nhớ trong vùng nhớ stack của hàm
- ☐ Truy cập vào các ô nhớ trong vùng nhớ stack bên ngoài hàm
- ☐ Truy cập vào vùng nhớ heap



Submit

#### Question #f2c54

1/1 point (graded)

Câu 8. Trường hợp nào sau đây không truy cập bộ nhớ an toàn về thời gian?

- ☒ Truy cập vào biến chưa được khởi tạo
- ☒ Truy cập vào vùng nhớ chưa được cấp phát
- ☐ Hàm không có kết quả trả về
- ☐ Hàm không có tham số vào



Submit

#### Question #3588f

1/1 point (graded)

Tiền điều kiện của hàm sau là gì?

```
1 void copyHead(char *src, char *dst, int n)
2 {
3     for(int i = 0; i < n; i++)
4         dst + i = src + i;
5 }
```

- ☒ `src != null`
- ☒ `dst != null`
- ☐ `n > 0`
- ☒ `n <= strlen(src)`
- ☒ `n <= strlen(dst)`
- ☐ `strlen(dst) >= strlen(src)`



