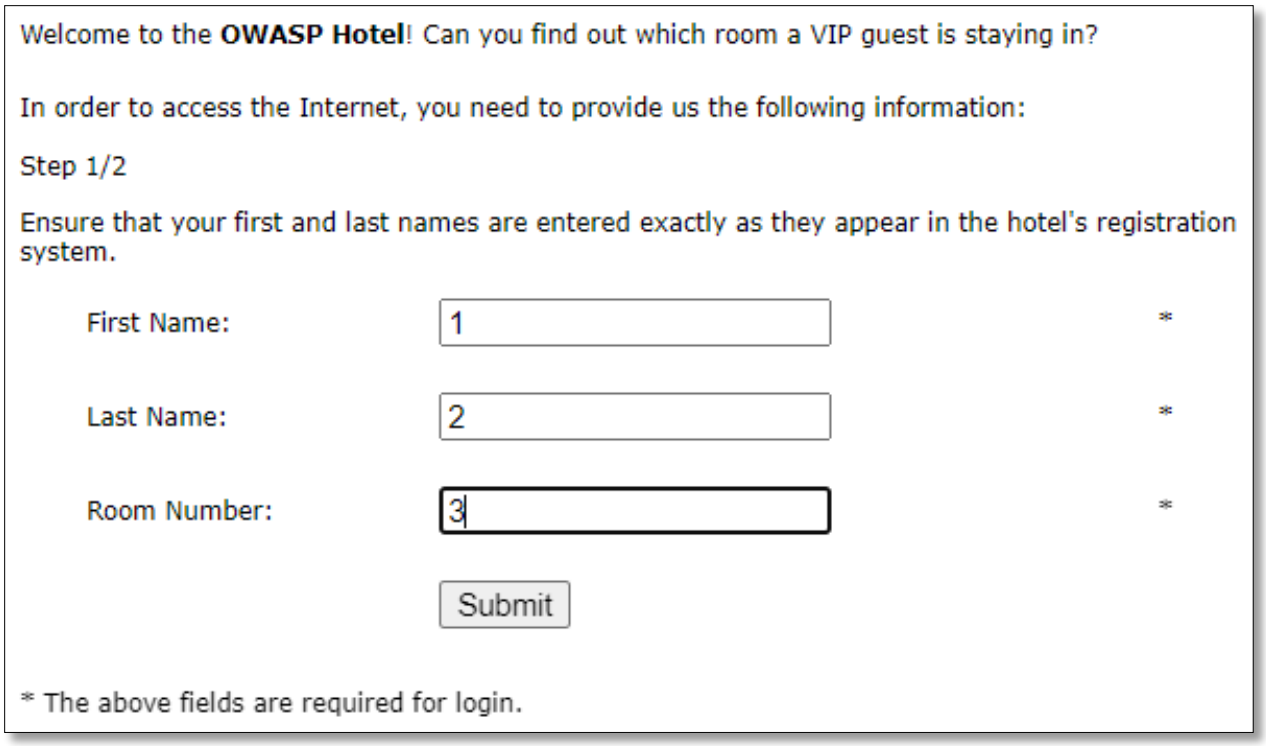
Buffer Overflows

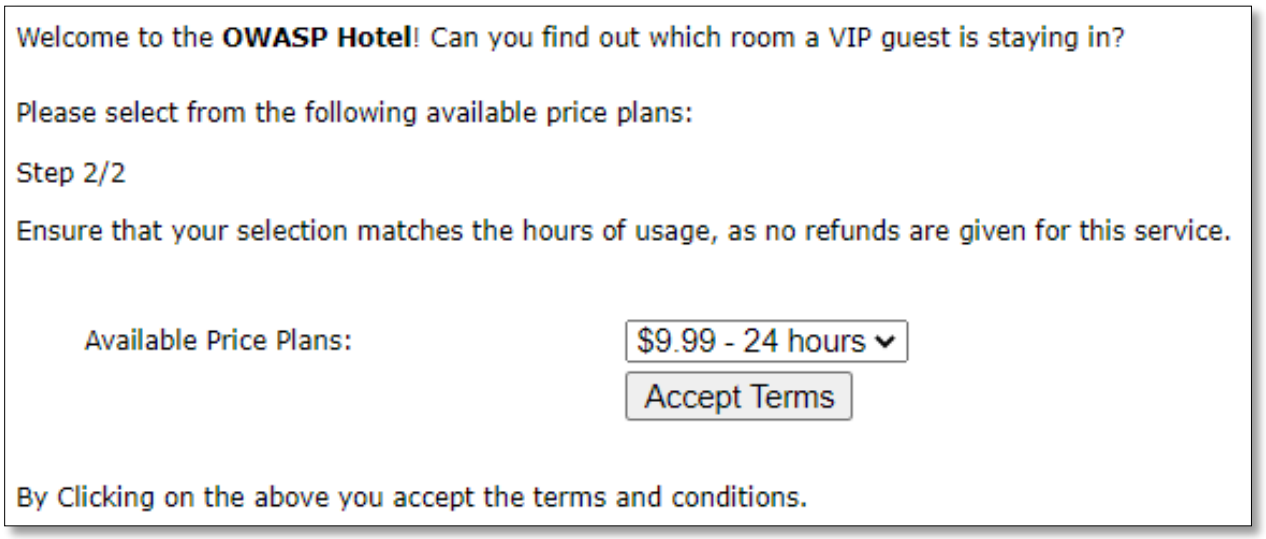
Tràn bộ nhớ đệm **"Buffer Overflows"**, là lỗi tràn bộ đệm của 1 vùng nhớ trong lập trình. Kiểu gì mà khi lập trình các bạn chả gặp lỗi này khi tương tác với mảng, vòng lặp, đệ quy, …

Mỗi chương trình khi thực thi sẽ được tải vào RAM, và sẽ được cung cấp một phần bộ nhớ, phần bộ nhớ này sẽ được chia ra làm nhiều loại, với mục đích khác nhau. Một phần sẽ được dùng để chứa mã lệnh, thường thì phần này sẽ chỉ để đọc và chạy các lệnh chứa trong đó, một phần khác lại được dùng để chứa dữ liệu, bạn có thể đọc và ghi thoải mái, nhưng phải xin cấp phát trước khi dùng.

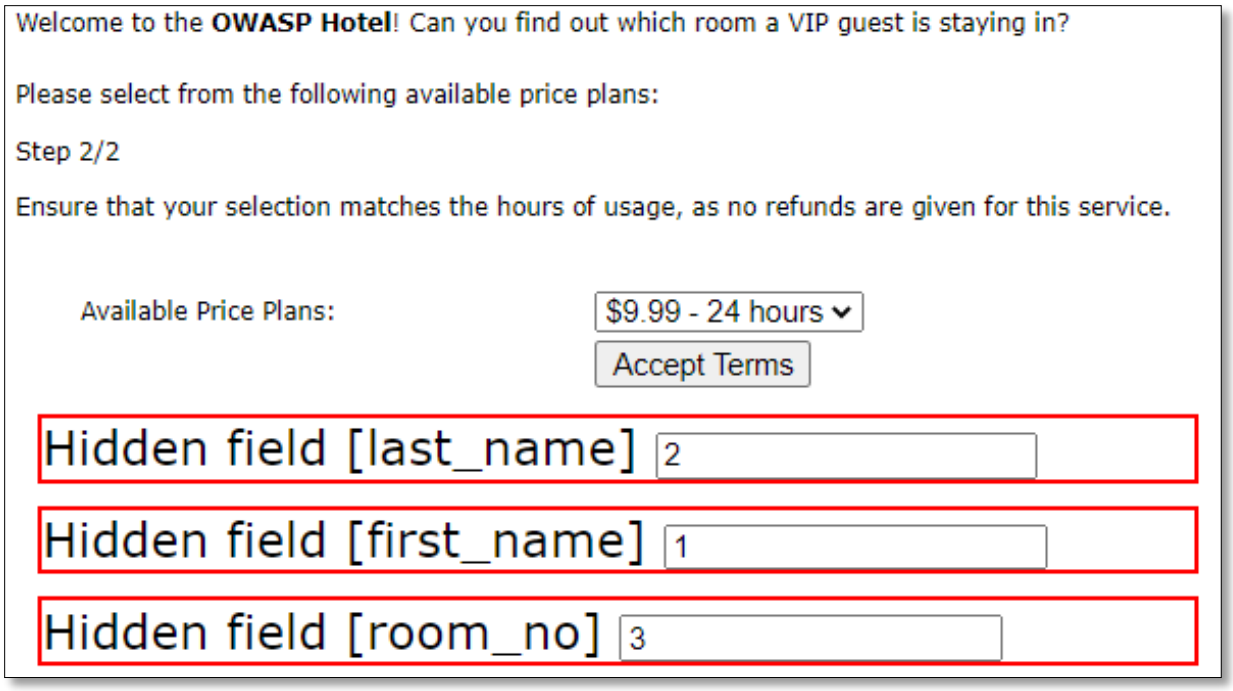
Thường khi tấn công kiểu này, người ta hay nhắm vào bộ nhớ STACK **"Stack Buffer Overflows"** của ứng dụng, một bộ nhớ LIFO (Last in First out),cái nào đẩy vào (PUSH) sau cùng thì sẽ được lấy ra (POP) đầu tiên. Nôm na là nếu bạn nắm được quy luật vận hành và cách thức tổ chức các ngăn xếp dữ liệu STACK của ứng dụng, và nó tồn tại lỗi tràn bộ đệm, bạn có thể chèn dữ liệu sang vùng nhớ khác bằng vùng nhớ bị tổn thương (bởi vì dữ liệu trong STACK khi bị tràn thì nó sẽ tràn lên vùng nhớ trước nó, và ghi đè dữ liệu của vùng nhớ đó).

Tấn công để kiểm tra lỗi này thì dễ, nhưng để nắm rõ về cách vận hành và khai thác hiệu quả của lỗi này thì ta cần có các kiến thức về khoa học máy tính.





Yêu cầu của bài này là tìm ra thông tin của người dùng phòng VIP. Thật ra sau khi mày mò cách làm trên mạng thì mình nhận thấy, có một số field đang ẩn trên website á, các field này cũng có thể thấy thông qua response khi bắt gói tin, bạn muốn theo dõi chúng thông qua đâu cũng được.



Okay, bài này lợi dụng việc tràn bộ đệm để tấn công. Nghĩ tới việc tấn công tràn bộ đệm là ta nghĩ ngay tới việc phải tạo ra 1 chuỗi dữ liệu dài thiệt dài để có thể phá hủy được không gian bộ nhớ được cấp phát, bạn muốn tạo dài bao nhiêu thì tạo, dài mà chưa đủ thì nhân đôi đoạn bạn tạo lên. Vì mình tạo ra 1 dãy số quá dài nên mình không thể copy vào đây được, nhưng nó là 1 dãy toàn số 1 thôi. Kiểu:

"1111111111111111111111111111111111111111111111111111….."

Vì lỗi tràn bộ đệm xuất hiện ở những nơi mà người dùng có thể nhập thông tin vào, thế nên ta ném thẳng cái chuỗi đó vào 3 vùng nhập luôn nhé.

Welcome to the **OWASP Hotel**! Can you find out which room a VIP guest is staying in?

In order to access the Internet, you need to provide us the following information:

Step 1/2

Ensure that your first and last names are entered exactly as they appear in the hotel's registration system.

First Name: 1111111111111111111111111 *

Last Name: 1111111111111111111111111 *

Room Number: 1111111111111111111111111 *

Submit

* The above fields are required for login.

Welcome to the **OWASP Hotel**! Can you find out which room a VIP guest is staying in?

Please select from the following available price plans:

Step 2/2

Ensure that your selection matches the hours of usage, as no refunds are given for this service.

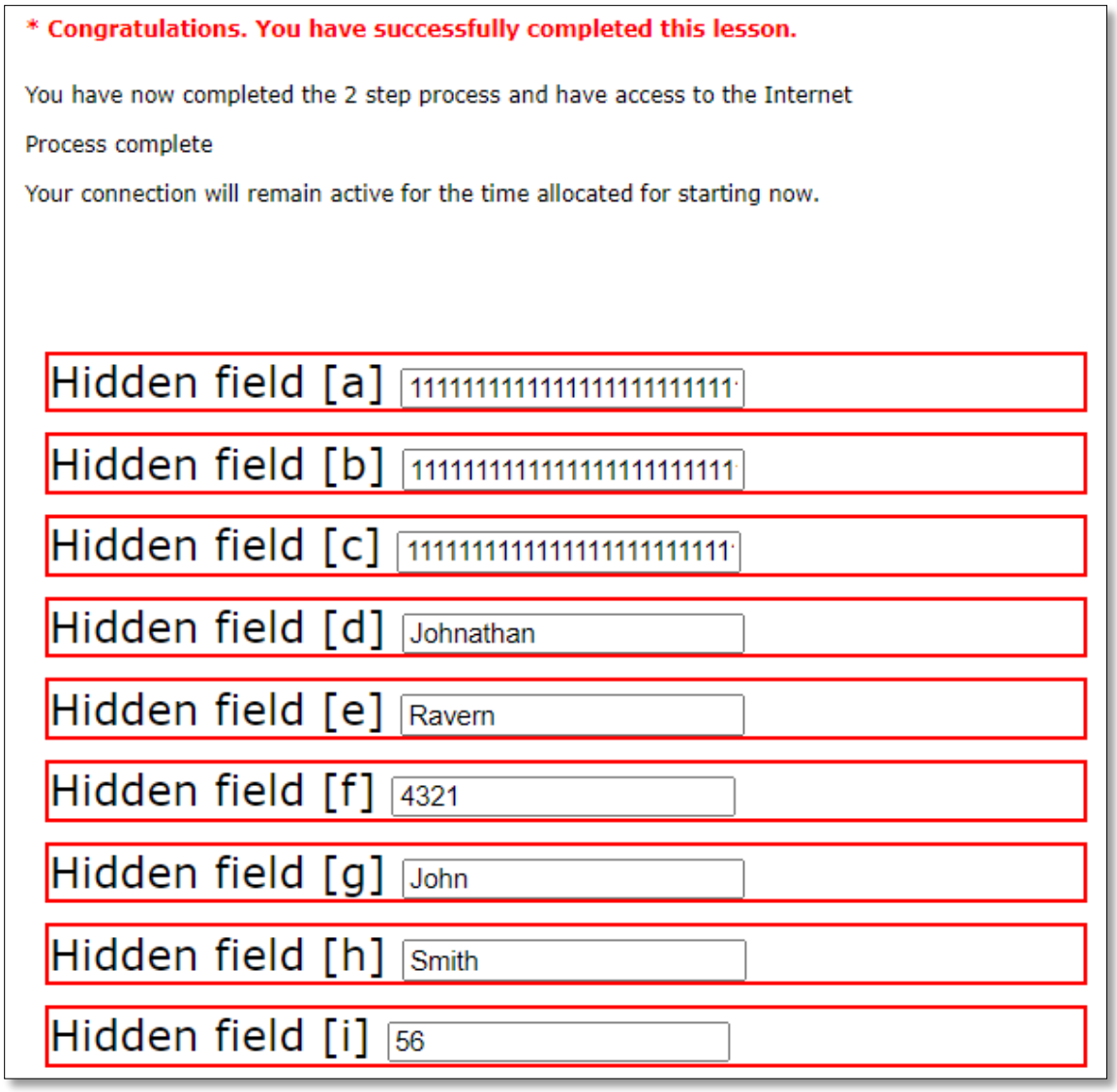
Available Price Plans: $9.99 - 24 hours

Accept Terms

Hidden field [last_name] 1111111111111111111111111

Hidden field [first_name] 1111111111111111111111111

Hidden field [room_no] 1111111111111111111111111



Okay, giờ đã có thể xem được thông tin của các người dùng VIP.

Mình đã làm vài bài về lỗi tràn bộ đệm, và đây không phải là cách khai thác đúng cho lỗi này. Nó giống như mô phỏng cho chúng ta cách mà lỗi có thể xảy ra thôi. Bởi vì lỗi tràn bộ đệm phần lớn là để khai thác trái phép vào các vùng nhớ khác, ghi đè và khiến nó hoạt động sai lệch.

```
// And finally the check...
if(param3.length() > 4096)
{
    ec.addElement(new Input(Input.hidden, "d", "Johnathan"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "e", "Ravern"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "f", "4321"));
    ec.addElement("\r\n");

    ec.addElement(new Input(Input.hidden, "g", "John"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "h", "Smith"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "i", "56"));
    ec.addElement("\r\n");

    ec.addElement(new Input(Input.hidden, "j", "Ana"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "k", "Arneta"));
    ec.addElement("\r\n");
    ec.addElement(new Input(Input.hidden, "l", "78"));
    ec.addElement("\r\n");
```

Và mình cũng tò mò, kiểm tra thử mã nguồn để xem nội dung bài tập thì mình thấy cái hàm này. Bài tập này chỉ kiểm tra liệu dữ liệu đầu vào của ứng dụng có lớn hơn 4096 bytes hay không, nếu có thì sẽ hiển thị các dữ liệu ẩn lên cho chúng ta.