BÁO CÁO THỰC HÀNH

**Môn học:** Lập trình an toàn & Khai thác lỗ hổng phần mềm

**Tên chủ đề:** Format String

*GVHD: Nguyễn Hữu Quyền*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT521.O12.ATCL

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Bùi Hoàng Trúc Anh | 21521817 | [21521817@gm.uit.edu.vn](mailto:21521817@gm.uit.edu.vn) |
| 2 | Nguyễn Ngọc Trà My | 21520353 | [215210353@gm.uit.edu.vn](mailto:215210353@gm.uit.edu.vn) |
| 3 | Lê Hoàng Oanh | 21521253 | [21521253@gm.uit.edu.vn](mailto:21521253@gm.uit.edu.vn) |
| 4 | Huỳnh Minh Tân Tiến | 21521520 | [21521520@gm.uit.edu.vn](mailto:21521520@gm.uit.edu.vn) |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-2)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Yêu cầu 1 | 100% |
| 2 | Yêu cầu 2 | 100% |
| 3 | Yêu cầu 3 | 100% |
| 4 | Yêu cầu 4 | 100% |
| 5 | Yêu cầu 5 | 100% |
| 6 | Yêu cầu 6 | 100% |
| 7 | Yêu cầu 7 | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

1. **Giả sử cần chuẩn bị chuỗi định dạng cho printf(). Sinh viên tìm hiểu và hoàn thành các chuỗi định dạng cần sử dụng để thực hiện các yêu cầu bên dưới.**

|  |  |
| --- | --- |
| **Yêu cầu** | **Chuỗi định dạng** |
| 1. In ra 1 số nguyên hệ thập phân | %d |
| 2. In ra 1 số nguyên 4 byte hệ thập lục phân, trong đó luôn in đủ 8 số hexan. | %08x |
| 3. In ra số nguyên dương, có ký hiệu + phía trước và chiếm ít nhất 5 ký tự, nếu không đủ thì thêm ký tự 0. | %+05d |
| 4. In tối đa chuỗi 8 ký tự, nếu dư sẽ cắt bớt. | %.8s |
| 5. In ra 1 số thực, trong đó đầu ra sẽ chiếm ít nhất 7 ký tự và luôn hiển thị 3 chữ số thập phân. Nếu số chữ số không đủ, nó sẽ đệm khoảng trắng ở phần nguyên. | %7.3f |
| 6. In ra 1 số thực, trong đó đầu ra sẽ chiếm ít nhất 7 ký tự và luôn hiển thị 3 chữ số thập phân. Nếu số chữ số không đủ, nó sẽ đệm ký tự 0 ở phần nguyên. | %07.3f |

1. **Sinh viên khai thác và truyền chuỗi s để đọc giá trị biến c của main. Giải thích ý nghĩa của chuỗi định dạng và lý do có thể in được giá trị cần thiết. Bonus: chuỗi s không dài hơn 10 ký tự.**

**Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình

Description automatically generated**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Description automatically generated

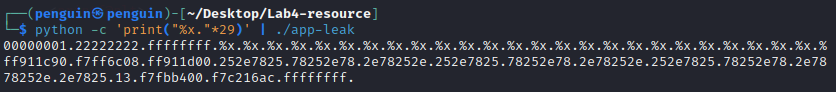
Như vậy, để đọc được đến dữ liệu tại khung màu xanh, cần bao 29 ký hiệu %x

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Description automatically generated

Chuỗi định dạng "%29$x" nói với hàm định dạng (ví dụ như printf) rằng nó nên lấy giá trị tại vị trí thứ 29 trên bảng xếp hạng và in nó dưới dạng số hexan.

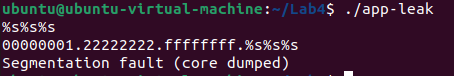
Lý do có thể in được giá trị cần thiết là do việc sử dụng lỗ hổng định dạng (format string vulnerability) trong chương trình. Nếu chương trình không kiểm soát đầu vào của chuỗi định dạng, tấn công như này có thể dẫn đến rò rỉ thông tin từ bộ nhớ.



So sánh giá trị k và m ở 2 cách này:

Trong cả hai cách đọc này, giá trị k và m đều là 29, và cả hai đang cố gắng in giá trị của tham số thứ 30 (29+1). Do đó, giá trị k và m giống nhau và in ra cùng một giá trị.

1. **Giải thích vì sao %s%s%s gây lỗi chương trình?**



A computer screen shot of a program

Description automatically generated

Đoạn code trên gây ra lỗi do việc sử dụng %s%s%s trong hàm printf(s). Khi đó, printf sẽ cố gắng tìm các biến tương ứng để in ra nhưng không thể tìm thấy, dẫn đến lỗi “Segmentation fault”.

Segmentation fault là một lỗi xảy ra khi chương trình cố gắng truy cập vào một vùng nhớ không được phép hoặc không tồn tại.

%s là một định dạng để in ra một chuỗi ký tự (string) từ một biến hoặc một hằng số. Khi sử dụng %s trong hàm printf, chương trình sẽ tìm kiếm một tham số có kiểu là char\* (con trỏ trỏ đến một chuỗi ký tự) để in ra.

%s%s%s là một chuỗi ký tự được nhập vào từ bàn phím thông qua hàm scanf("%s", s). Khi đó, biến s sẽ lưu trữ chuỗi ký tự này. Khi sử dụng printf(s), chương trình sẽ in ra nội dung của biến s, tức là %s%s%s.

Tuy nhiên, khi sử dụng printf(s), chương trình cũng sẽ hiểu rằng s là một định dạng để in ra các biến khác, và sẽ tìm kiếm các biến có kiểu là char\* để in ra theo định dạng %s%s%s. Nhưng do không có biến nào được truyền vào hàm printf, chương trình sẽ truy cập vào các vùng nhớ ngẫu nhiên, dẫn đến lỗi “Segmentation fault”.  
  
**Sinh viên khai thác và truyền chuỗi s đọc thông tin từ Global Offset Table (GOT) và lấy về địa chỉ của hàm scanf. Giải thích ý nghĩa của chuỗi định dạng và lý do có thể in được giá trị cần thiết.**

**A computer screen shot of a program

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

Chuỗi định dạng %p%p%p có mục đích là in ra vị trí của hàm scanf.  
  
Chuỗi định dạng %p%p%p sẽ đọc ba địa chỉ từ GOT. Địa chỉ đầu tiên sẽ là địa chỉ của hàm scanf. Đây là địa chỉ mà chúng ta cần lấy được.  
Khi chương trình nhận được chuỗi định dạng, nó sẽ thực hiện thao tác định dạng tương ứng. Trong trường hợp này, thao tác định dạng sẽ là in ra địa chỉ của tham số thứ nhất.  
Tham số thứ nhất của thao tác định dạng là địa chỉ của hàm scanf. Do đó, chương trình sẽ in ra địa chỉ của hàm scanf.

Sinh viên giải thích ý nghĩa dòng code 16 để lấy địa chỉ của scanf từ GOT?  
Code này sử dụng hàm u32() để chuyển đổi 4 byte đầu tiên của phản hồi từ hàm recv() thành một số nguyên 32 bit. Số nguyên này sau đó được chuyển đổi thành định dạng hex bằng hàm hex(). Kết quả cuối cùng được in ra màn hình.  
Trường hợp này, hàm recv() được sử dụng để nhận phản hồi từ ứng dụng. Phản hồi này bao gồm địa chỉ của hàm scanf trong bộ nhớ. Dòng code trên sẽ in ra địa chỉ này dưới dạng định dạng hex. sh.recv()[4:8] dùng để truy cập các byte đầu tiên của phản hồi từ hàm recv().  
+) sh.recv(): Nhận dữ liệu từ quá trình con (child process) thông qua đối tượng sh, là quá trình đã được khởi chạy bằng process('./app-leak').  
+) [4:8]: Lấy một phần của dữ liệu nhận được, từ byte thứ 4 đến byte thứ 7. +) u32(): Chuyển đổi một chuỗi 4 byte thành một số nguyên 32-bit. Trong trường hợp này, chuyển đổi địa chỉ của hàm scanf từ dạng chuỗi thành dạng số nguyên. +) hex(): Chuyển đổi số nguyên thành chuỗi hex

1. **Sinh viên khai thác và truyền chuỗi s để ghi đè biến c của file app-overwrite thành giá trị 16. Giải thích ý nghĩa của chuỗi định dạng và lý do có thể in được giá trị cần thiết.**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Description automatically generated

*Địa chỉ của biến c.*

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, màn hình

Description automatically generated

Địa chỉ của chuỗi s

Ảnh có chứa văn bản, ảnh chụp màn hình, màn hình, phần mềm

Description automatically generated

*Địa chỉ tham số đầu tiên của hàm printf*

Ảnh có chứa văn bản, ảnh chụp màn hình, thực đơn, Phông chữ

Description automatically generated

Biết địa chỉ của chuỗi s là 0xffffcf28 và địa chỉ tham số của hàm printf bắt đầu từ vị trí 0xffffcf10. Do đó, vị trí lưu chuỗi định dạng s sẽ tương ứng với tham số thứ 7 của printf.

Để in được 12 ký tự, có thể sử dụng: %12d hoặc %12s trong chuỗi định dạng. Điều này sẽ yêu cầu hàm printf in ra một số nguyên hoặc một chuỗi với độ dài cố định là 12 ký tự.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Description automatically generated

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Description automatically generated

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Description automatically generated

1. **Sinh viên khai thác và truyền chuỗi s để ghi đè biến a của file app-overwrite thành giá trị 2. Giải thích ý nghĩa của chuỗi định dạng và lý do có thể in được giá trị cần thiết.**

Additional format là “aa” sẽ chiếm 2 byte, “%k$n” chiếm 4 byte như vậy còn cần padding thêm 2 byte để đẩy vị trí lưu địa chỉ của biến a được bắt đầu tại vị trí địa chỉ chia hết cho 4. Do đó vị trí của k sẽ là 8.

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, phần mềm

Description automatically generated**

Ảnh có chứa văn bản, Phông chữ, phần mềm, Phần mềm đa phương tiện

Description automatically generated

1. **Sinh viên khai thác và truyền chuỗi s để ghi đè biến b của file app-overwrite thành giá trị 0x12345678. Báo cáo chi tiết các bước phân tích, xác định chuỗi định dạng và kết quả khai thác.**

0x12345678 = 305419896, nếu buff vào 1 ô nhớ bằng %n, thì phải in thành công được 305419896 ký tự, 1 số khá lớn cho việc in ra để %n có thể ghi nhận, nên thay vì chỉ buff vào 1 ô địa chỉ, sẽ tiến hành buff thành các ô nhớ liên tục theo little endian.

Đầu tiên xác định địa chỉ của biến b

A screen shot of a computer code

Description automatically generated

ta cần phải làm là ghi đè lên từng địa chỉ này

* 0x0804a028 - Đây sẽ là địa chỉ overwrite   
  Byte tại địa chỉ **0x0804a028**: **0x78**
* Byte tại địa chỉ **0x0804a028 + 1**: **0x56**
* Byte tại địa chỉ **0x0804a028 + 2**: **0x34**
* Byte tại địa chỉ **0x0804a028 + 3**: **0x12**

A screenshot of a computer screen

Description automatically generated  
Xác định được tham số của chuỗi S trong hàm printf() là 7 , ta sẽ cần phải ghi đè bắt đầu từ %6$n   
  
overwrite address : b\_addr , b\_addr+1, b\_addr+2, b\_addr+3,   
index pram: A%6$n, B%7$n, C%8$n, D%9$n  
padding:   
A: %n phải ghi nhận được 0x78= 120 ký tự được in ra, do overaddr được ghi dạng 32, được hiểu là thành 4 ký tự với mỗi địa chỉ, nên tổng là 16, cần padding them 104  
B: phải ghi nhận được 0x56 = 86 ký tự, nhưng do là đây là các địa chỉ liên tiếp, nên B cũng sẽ ghi nhận luôn giá trị của A, nên giá trị của B mặc định đã là 120 + , thì không thể nào ghi nhận được giá trị 0x56, nên phải ghi nhận 0x156, do có kích thước 1 byte nên lúc tính toán thì cũng sẽ vẫn là 0x56 ( 1 bị loại bỏ do overflow) : 0x156 = 342, padding = 342 – 120( của A) = 222  
C: tương tự, phải ghi nhận giá trị 0x34 = 52 , nhưng mặc định C đã có giá trị là 342+   
nên phải ghi nhận 0x234, padding sẽ là : 222

D: tượng tự, phải ghi nhận giá trị 0x312, padding là 222

A computer screen shot of numbers and symbols

Description automatically generated

A screen shot of a computer

Description automatically generated

Các bước thực hiện/ Phương pháp thực hiện/Nội dung tìm hiểu (Ảnh chụp màn hình, có giải thích)

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên nộp bài theo thời gian quy định trên course.

**Báo cáo:**

* File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
* Đặt tên theo định dạng: **[Mã lớp]-AssignmentX\_NhomY** (trong đó X là Thứ tự Assignment, Y là số thứ tự nhóm đồ án theo danh sách đã đăng ký).

*Ví dụ: [NT521.O11.ANTT]-Assignment01\_Nhom03.pdf.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.

**Đánh giá**:

* Hoàn thành tốt yêu cầu được giao.
* Có nội dung mở rộng, ứng dụng.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-2)