

Giải thích từng phần:

**O:4:"User":2:**

O là viết tắt của Object (đối tượng).

4 là độ dài của tên lớp đối tượng (ở đây là "User", có 4 ký tự).

"User" là tên của lớp đối tượng (class).

2 là số thuộc tính (property) mà đối tượng này có.

**s:4:"name";s:5:"Nhom2";:**

s là viết tắt của String (chuỗi).

4 là độ dài của chuỗi tên thuộc tính (ở đây là "name", có 4 ký tự).

"name" là tên của thuộc tính đầu tiên.

5 là độ dài của giá trị chuỗi (ở đây là "Nhom2", có 5 ký tự).

"Nhom2" là giá trị của thuộc tính name.

**s:10:"isLoggedIn";b:1;:**

s:10 là chuỗi "isLoggedIn" có độ dài 10 ký tự.

b:1 là giá trị của thuộc tính isLoggedIn, với b đại diện cho Boolean và 1 có nghĩa là true (đăng nhập thành công).

Tóm lại, đoạn này biểu diễn một đối tượng User có hai thuộc tính:

name với giá trị "Nhom2".

isLoggedIn với giá trị true.

Giải thích lý do vì sao chạy vulnerable-app-1.php in được name của class NormalClass?

Đầu tiên ta nhìn vào file classes.php và vulnerable-app-1.php

A computer screen shot of white text

Description automatically generated

A computer screen shot of white text

Description automatically generated

Ta thấy chức năng của file vulnerable-app-1.php là thực hiện việc deserialize nội dung của file serial\_Nhom2 trở thành dạng Object của class NormalClass và khi Object bị hủy thì sẽ thực hiện hàm hủy của class NormalClass là ghi ra thuộc tính name ra màn hình.

A close up of text

Description automatically generated

Kết quả thực hiện:

A screenshot of a computer

Description automatically generated

Kết quả có sự khác biệt là thay vì in ra name của class NormalClass thì lần này in ra kết quả của lệnh “ls”. Lý do cho việc này chính là ở việc ta thay đổi NormalClass sang DangerousClass ta thấy ở class DangerousClass thì khi thực hiện hàm hủy thì sẽ thực hiện lệnh echo passthru($this->cmd); (Hàm passthru() trong PHP dùng để **thực thi một lệnh hệ thống** và trực tiếp in kết quả của lệnh đó ra màn hình mà không cần phải trả về kết quả.). Thuộc tính cmd chính là lệnh “ls” đã được định nghĩa trước đó trong hàm tạo của class DangerousClass

A black background with white text

Description automatically generated



Chức năng của DangerousClass có thể bị khai thác vì nó sử dụng hàm passthru() trong phương thức \_\_destruct(). Đây là một vấn đề bảo mật rất nghiêm trọng do hàm passthru() cho phép thực thi lệnh hệ thống mà không kiểm tra kỹ các đầu vào. Cụ thể:

Nguy cơ khai thác:

A computer screen shot of white text

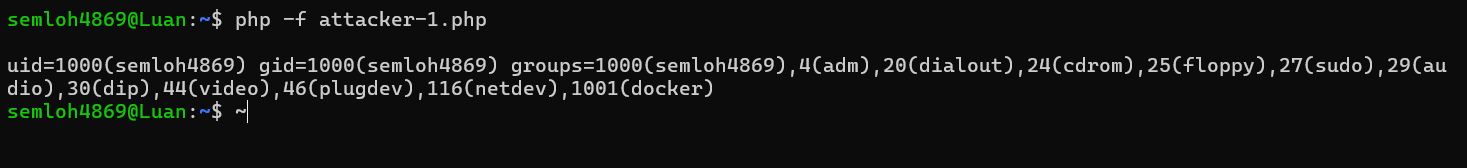
Description automatically generated

Hàm passthru():

Hàm này thực thi lệnh hệ thống (trong ví dụ trên là lệnh ls để liệt kê các tệp và thư mục) và in kết quả ra màn hình.

Nếu một kẻ tấn công có thể thao tác hoặc thay đổi thuộc tính $cmd của đối tượng DangerousClass thành một lệnh hệ thống nguy hiểm, họ có thể thực thi bất kỳ lệnh hệ thống nào trên máy chủ.





Code:

A computer screen shot of white text

Description automatically generated