

## **Experiment 6**

**Aim:** Use burp proxy to test web applications

### **Theory:**

#### **What is burp suite**

Burp Suite is a popular cybersecurity tool used for web application security testing and assessment. It is developed by PortSwigger, a company that specializes in web application security. Burp Suite is widely used by security professionals, penetration testers, and developers to identify and mitigate security vulnerabilities in web applications.

#### **Features:**

Burp Suite is a comprehensive web application security testing tool that offers a wide range of features to help security professionals, developers, and penetration testers identify and mitigate security vulnerabilities. Some of the key features of Burp Suite include:

##### **Intercept everything your browser sees**

Burp Suite's built-in browser works right out of the box - enabling you to modify every HTTP message that passes through it.

##### **Quickly assess your target**

Determine the size of your target application. Auto-enumeration of static and dynamic URLs, and URL parameters.

##### **Speed up granular workflows**

Modify and reissue individual HTTP and WebSocket messages, and analyze the response - within a single window.

##### **Manage recon data**

All target data is aggregated and stored in a target site map - with filtering and annotation functions.

##### **Expose hidden attack surface**

Find hidden target functionality with an advanced automatic discovery function for "invisible" content.

##### **Break HTTPS effectively**

Proxy even secures HTTPS traffic, using Burp Suite's built-in instrumented browser.

##### **Work with HTTP/2**

Burp Suite offers unrivaled support for HTTP/2-based testing - enabling you to work with HTTP/2 requests in ways that other tools cannot.

## **Work with WebSockets**

WebSockets messages get their own specific history - allowing you to view and modify them.

## **Manually test for out-of-band vulnerabilities**

Make use of a dedicated client to incorporate Burp Suite's out-of-band (OAST) capabilities during manual testing.

## **DOM Invader**

Use Burp Suite's built-in browser to test for DOM XSS vulnerabilities more easily - with DOM Invader.

## **Assess token strength**

Easily test the quality of randomness in data items intended to be unpredictable (e.g. tokens).

## **Tools offered by burp suite:**

**Proxy:** Burp Suite's Proxy module allows you to intercept and modify HTTP and HTTPS traffic between your browser and the target web application. This feature is essential for understanding and manipulating requests and responses, which helps in identifying potential vulnerabilities.

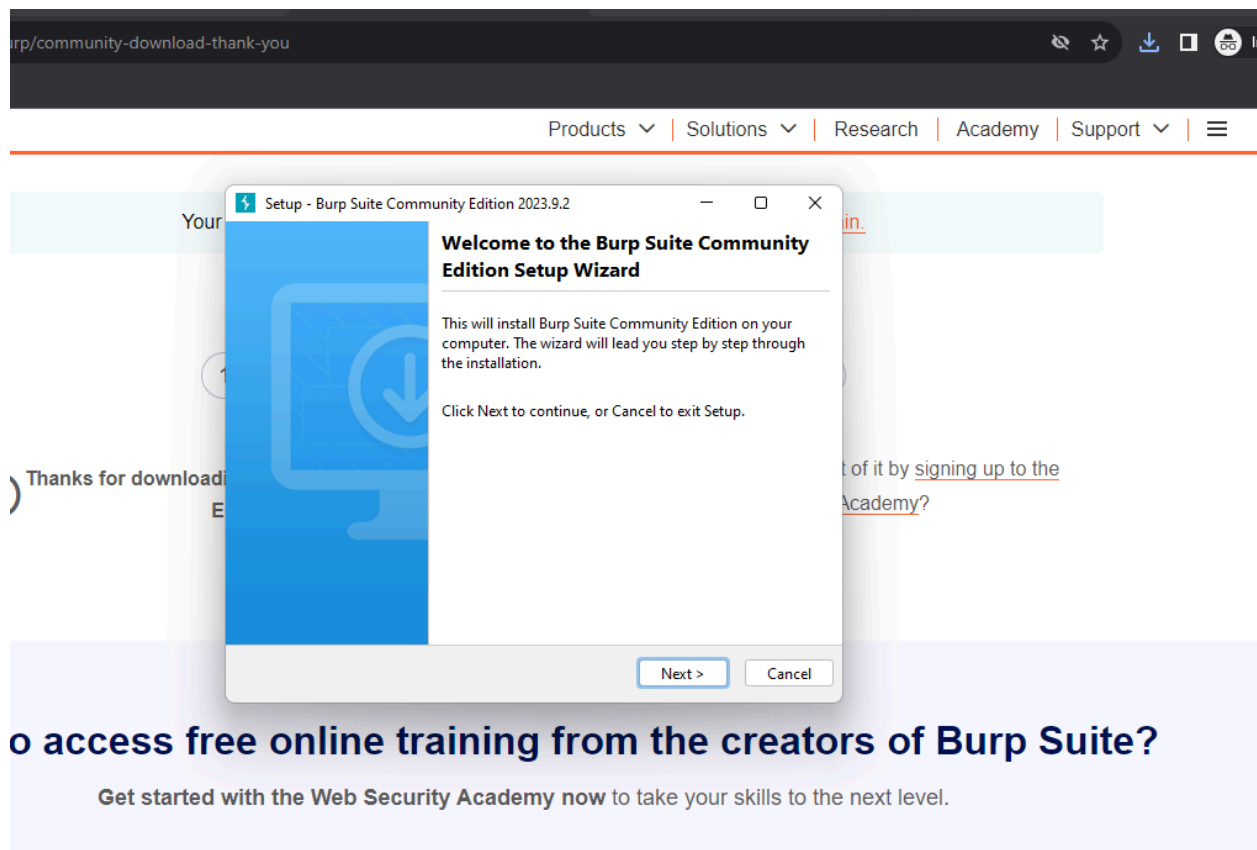
**Scanner:** The Scanner module automates the process of identifying security vulnerabilities in web applications. It can detect a variety of vulnerabilities, including SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and more.

**Spider:** The Spider module crawls through the web application, following links and mapping out the application's structure. It helps in identifying all accessible parts of the application, which is crucial for effective testing.

**Intruder:** The Intruder module is used for performing automated attacks against web applications, including brute-force attacks, fuzzing, and payload testing. It allows you to test different input combinations and identify weak points in the application.

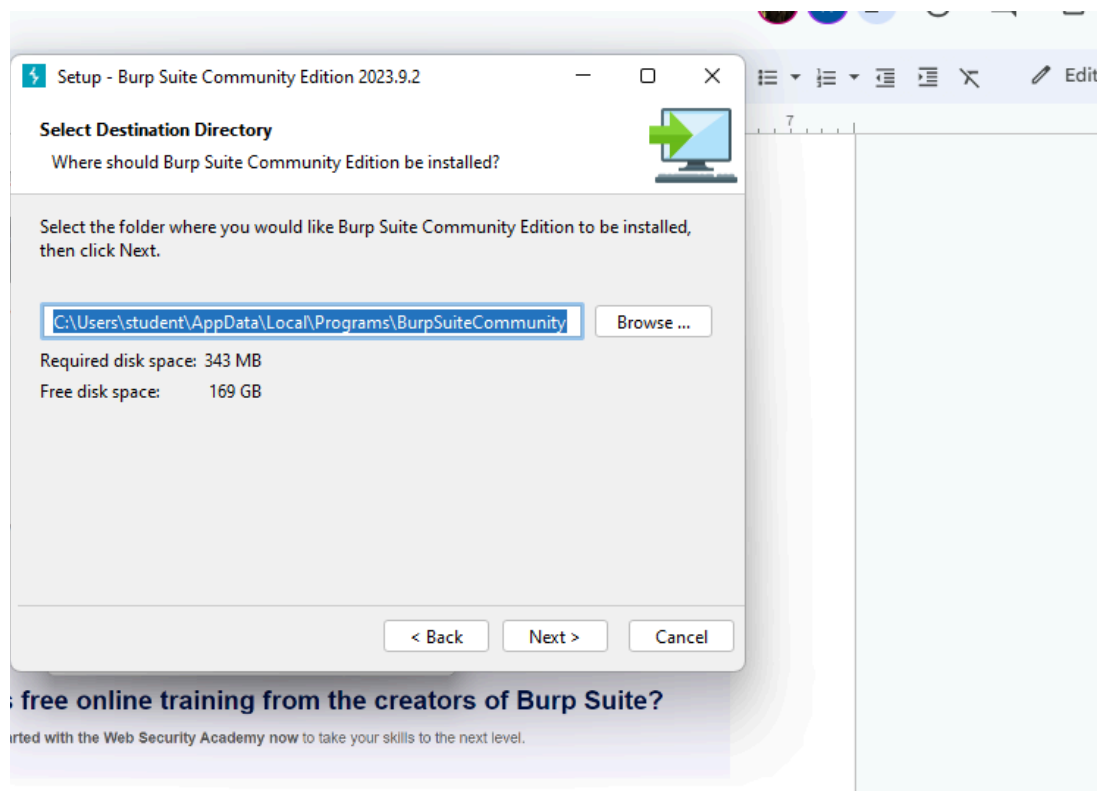
**Repeater:** The Repeater module enables you to manually modify and replay individual requests. This feature is useful for testing specific scenarios and observing how the application responds to different inputs.

Steps involved in Downloading:



o access free online training from the creators of Burp Suite?

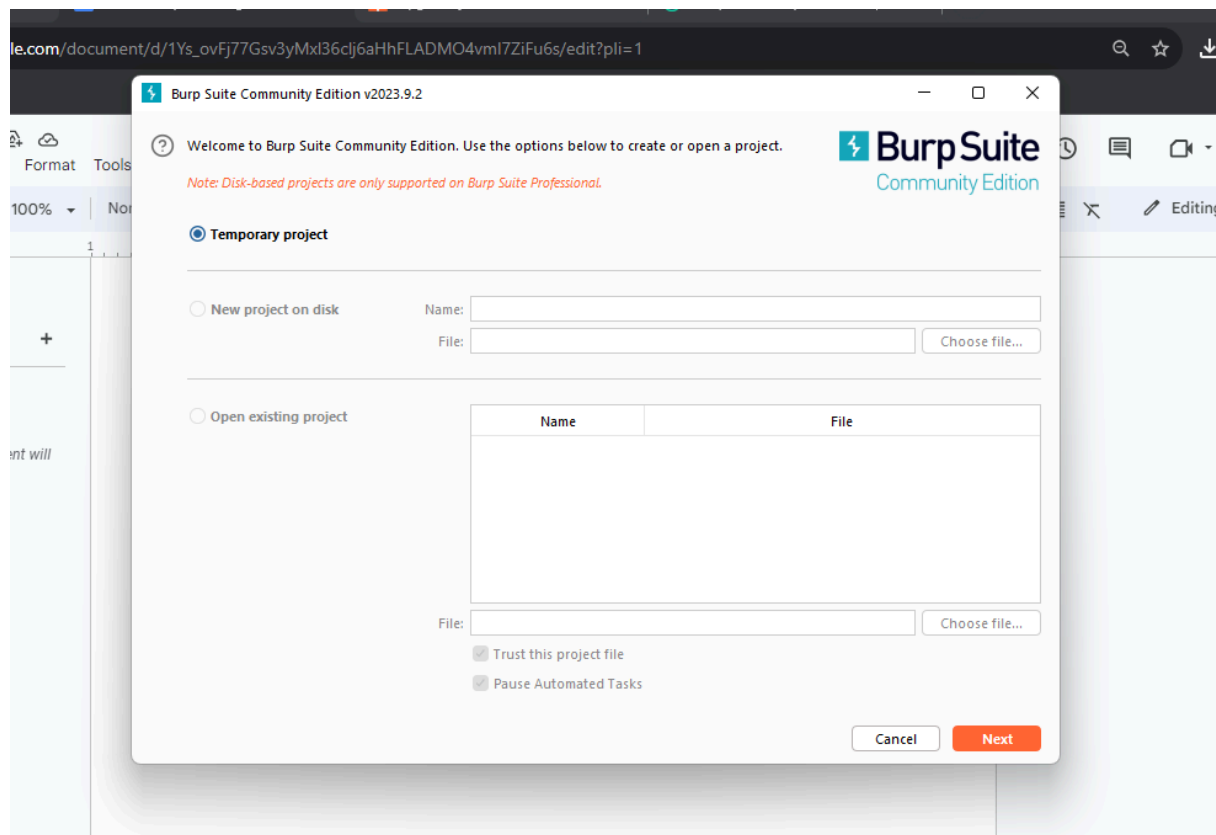
Get started with the Web Security Academy now to take your skills to the next level.



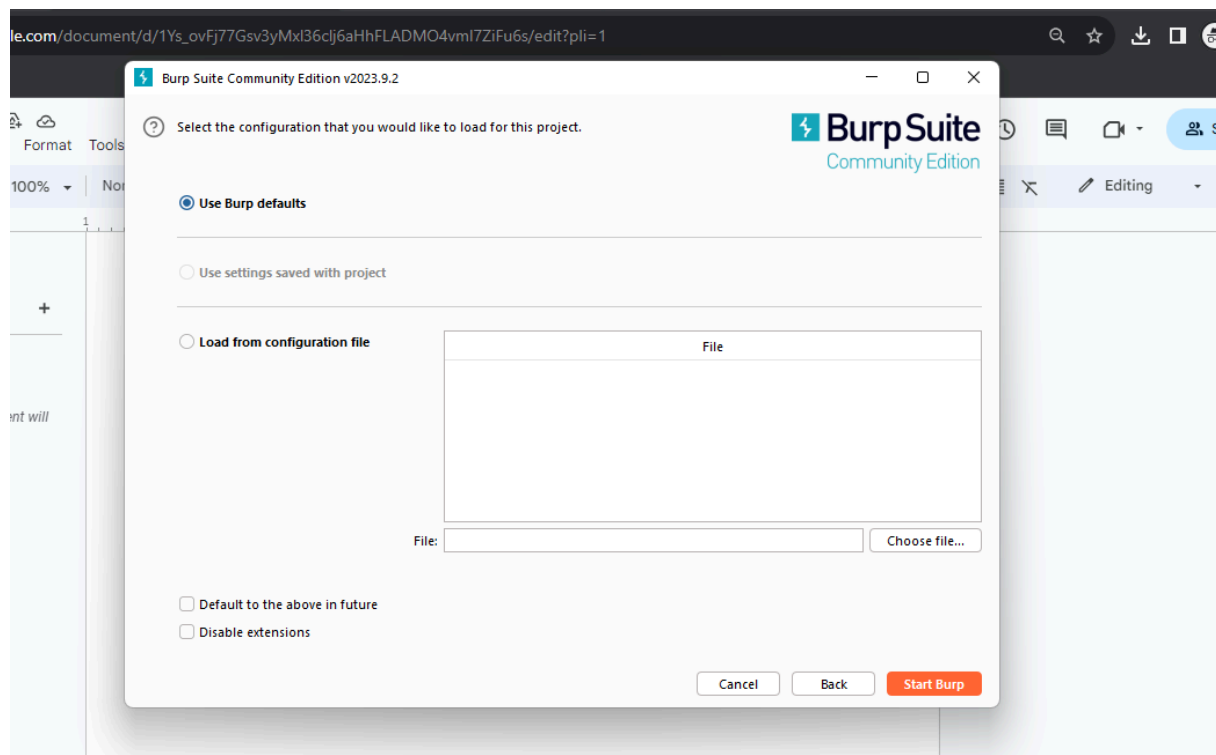
free online training from the creators of Burp Suite?

Get started with the Web Security Academy now to take your skills to the next level.

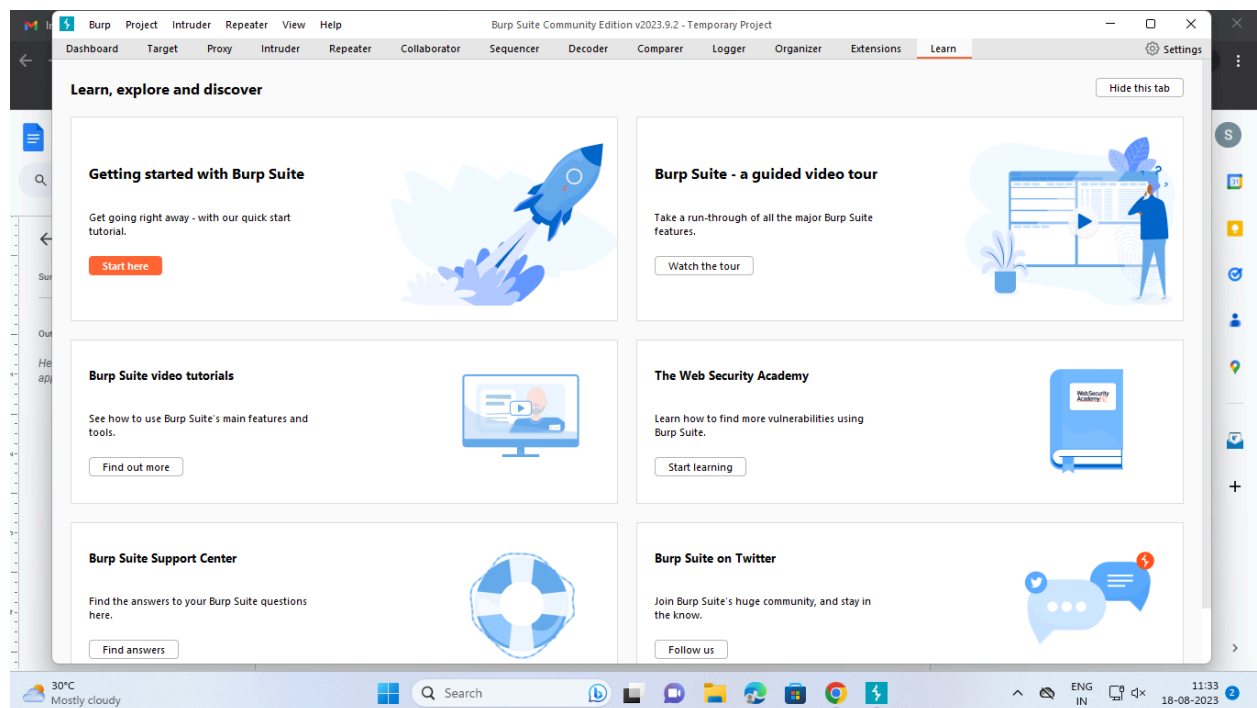
Create a new temporary project or select an existing project



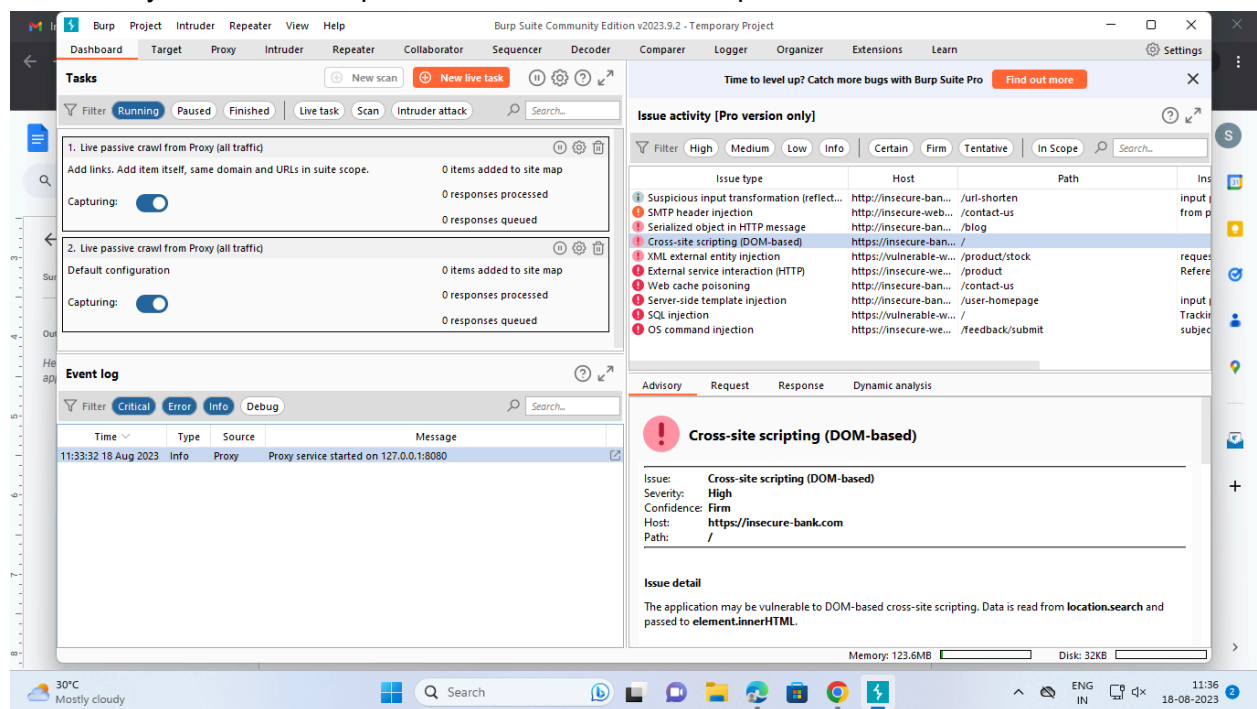
Click on start burp after either loading default configurations or loading configurations file



Click on dashboard to see all the statistics

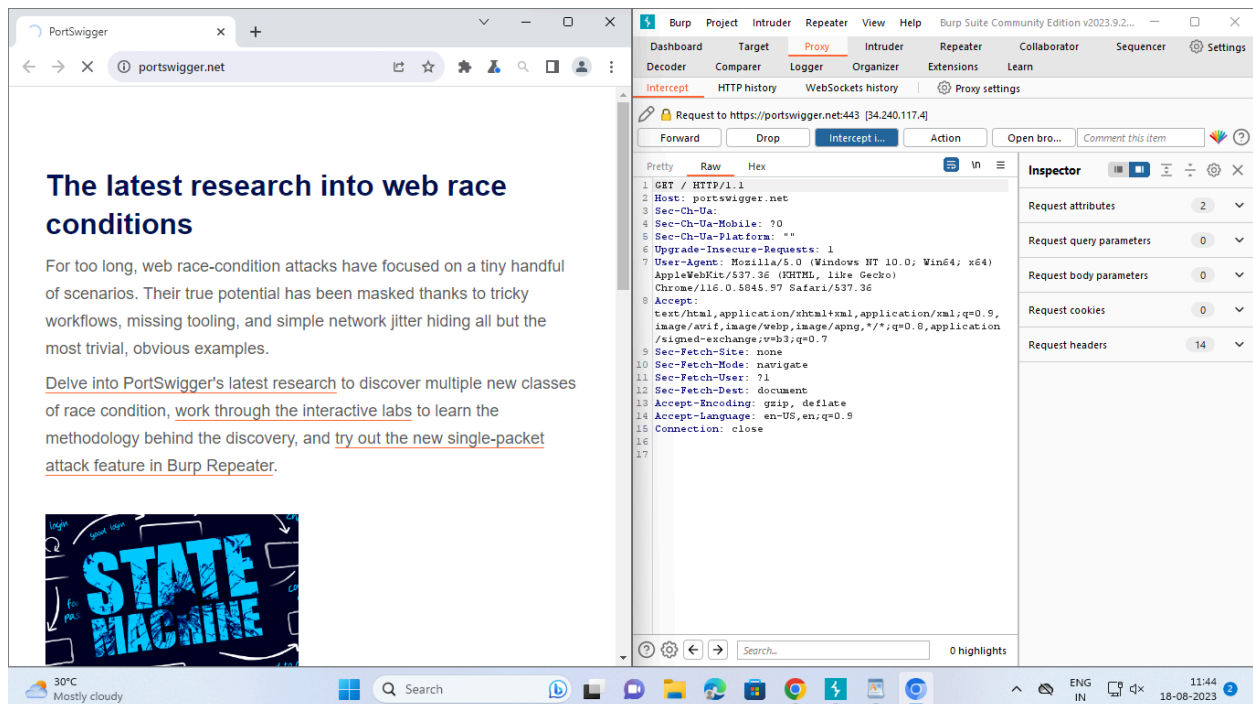


Go to Proxy -> Turn Intercept from off to on -> click on Open browser

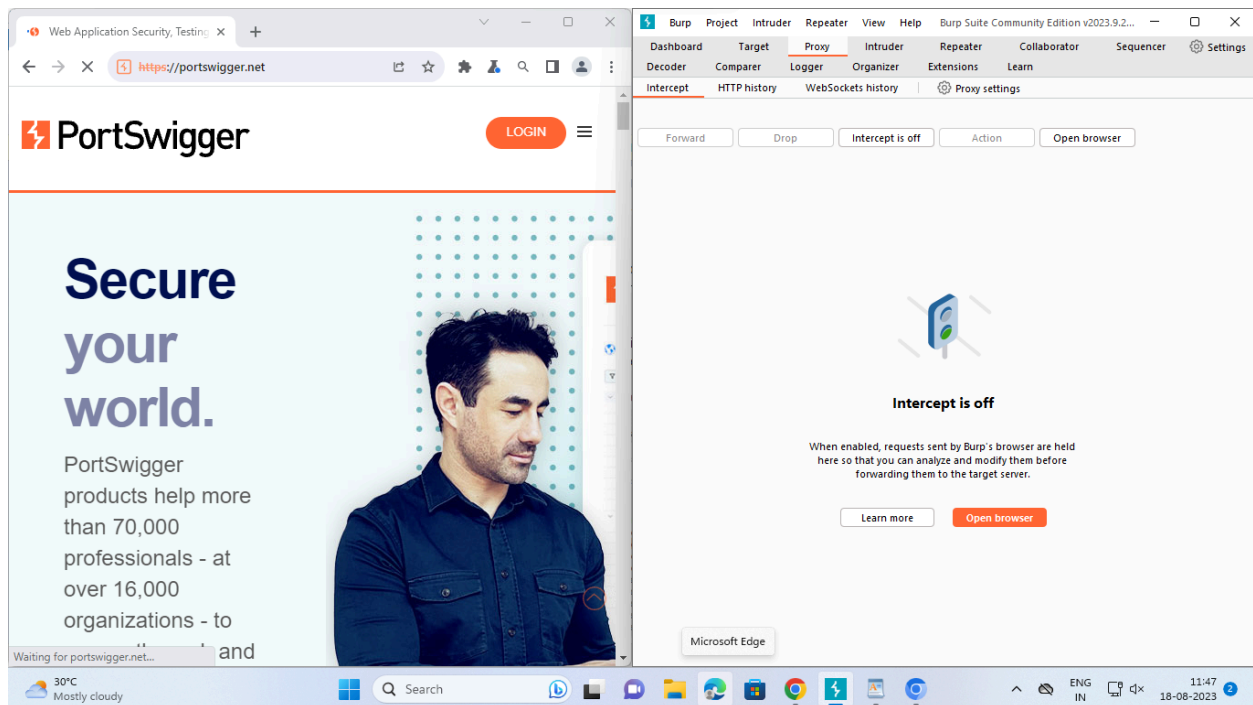


Open <https://portswigger.net/>

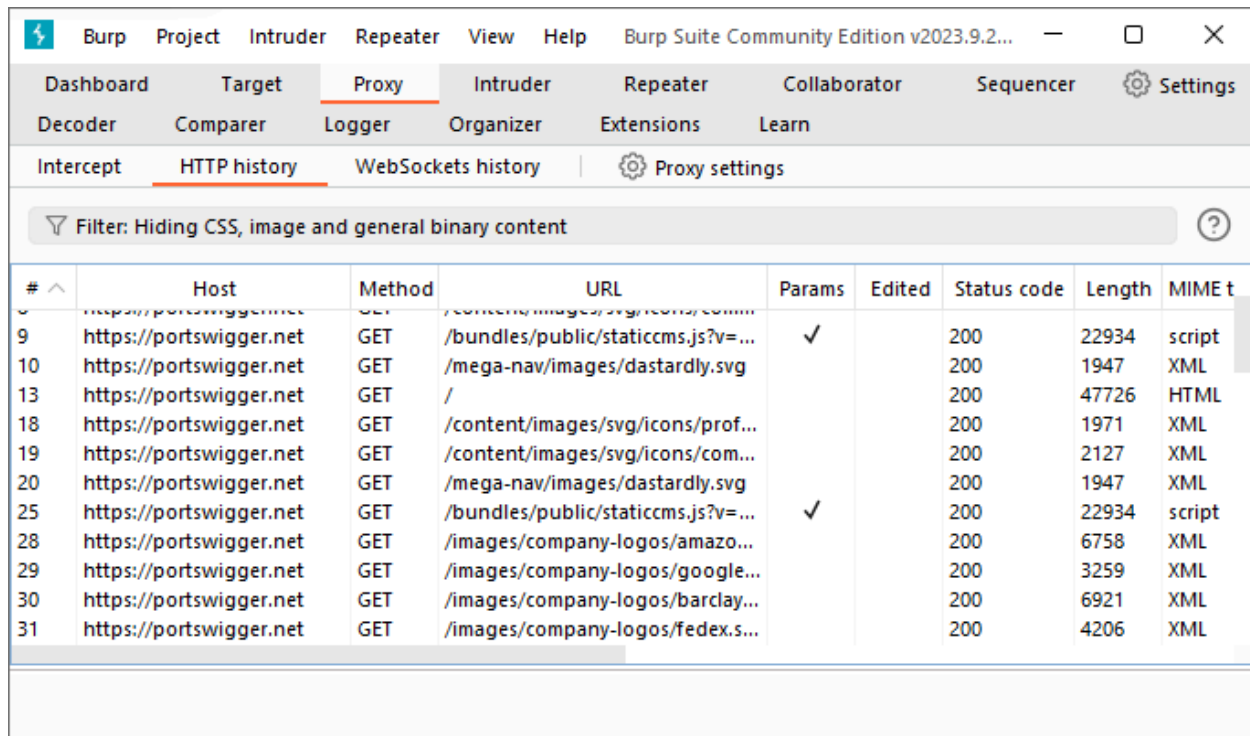
As you can see the site keeps loading as the request has now been intercepted. We can now analyze and then forward the request to the server to load the page



Interception is done and the site has been forwarded to the server and the site has loaded in the user side.



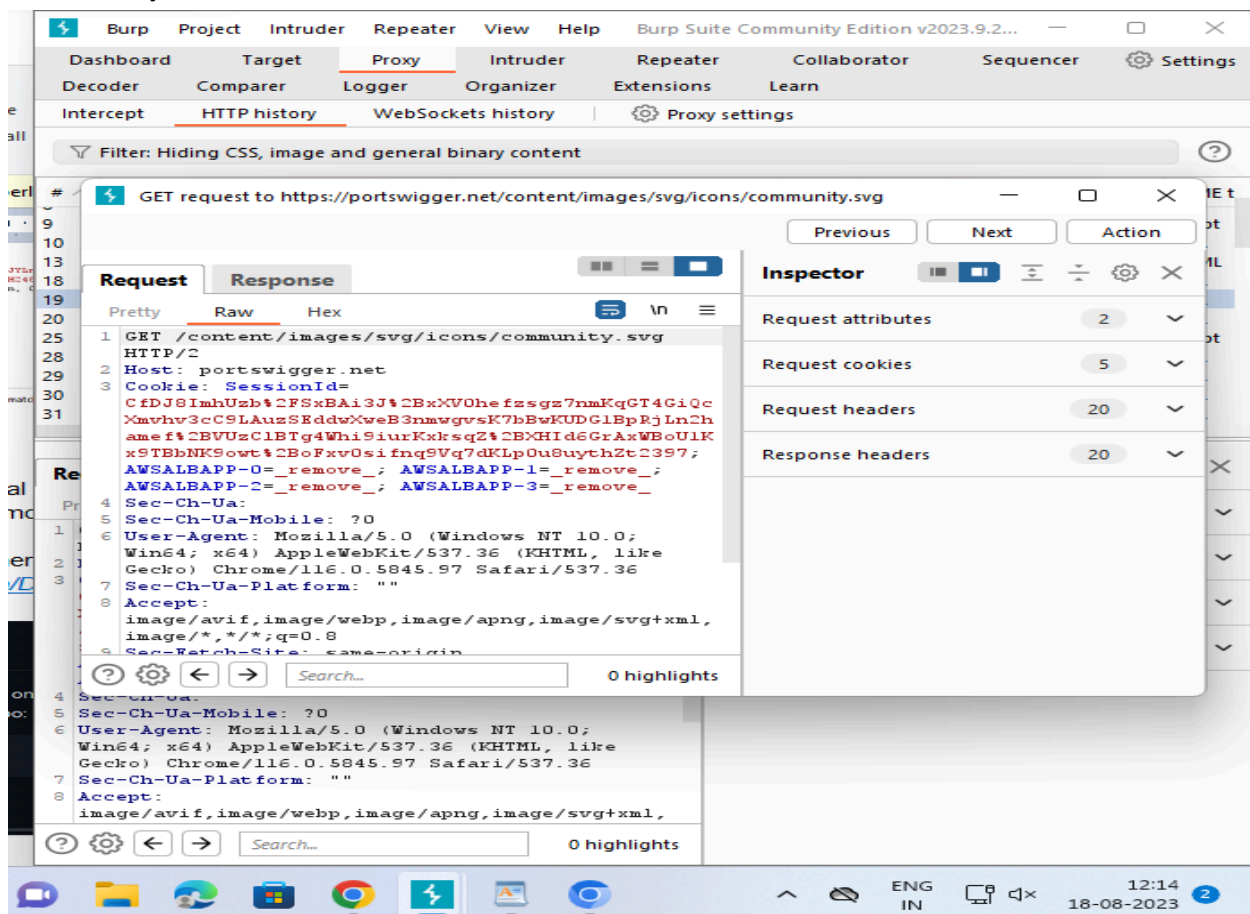
Go to HTTP History , where we can see all the logs of the user has been to even when intercept is off



The screenshot shows the Burp Suite interface with the 'HTTP history' tab selected. A filter is applied: 'Hiding CSS, image and general binary content'. The table below lists the intercepted requests.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME t
9	https://portswigger.net	GET	/bundles/public/staticcms.js?v=...	✓		200	22934	script
10	https://portswigger.net	GET	/mega-nav/images/dastardly.svg			200	1947	XML
13	https://portswigger.net	GET	/			200	47726	HTML
18	https://portswigger.net	GET	/content/images/svg/icons/prof...			200	1971	XML
19	https://portswigger.net	GET	/content/images/svg/icons/com...			200	2127	XML
20	https://portswigger.net	GET	/mega-nav/images/dastardly.svg			200	1947	XML
25	https://portswigger.net	GET	/bundles/public/staticcms.js?v=...	✓		200	22934	script
28	https://portswigger.net	GET	/images/company-logos/amazo...			200	6758	XML
29	https://portswigger.net	GET	/images/company-logos/google...			200	3259	XML
30	https://portswigger.net	GET	/images/company-logos/barclay...			200	6921	XML
31	https://portswigger.net	GET	/images/company-logos/fedex.s...			200	4206	XML

Click on any one of the sites to check details about it and the headers,cookies ,etc



The screenshot shows the Burp Suite interface with the 'HTTP history' tab selected. A filter is applied: 'Hiding CSS, image and general binary content'. The details of a specific request are shown in the 'Inspector' pane.

Request: GET /content/images/svg/icons/community.svg

Host: portswigger.net

Cookie: SessionId=CfDJ8ImhUzb%2FSxBAi3J%2BxXV0hefzsgz7nmKqGT4GiQcXmVhv3cC9LAuzSEddwXweB3nmwgvsk7bBwKUDG1BpRjLn2hamef%2BVUzC1BTg4Whi9iurKxksq2%2BCHI d6GrAxWB0U1Kx9TBbNKsowt%2BoFxxv0sifnqSVq7dKLP0u8uythZtC397; AWSALBAPP-0=\_remove\_; AWSALBAPP-1=\_remove\_; AWSALBAPP-2=\_remove\_; AWSALBAPP-3=\_remove\_

Sec-Ch-Ua: Sec-Ch-Ua-Mobile: ?0

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.5845.97 Safari/537.36

Sec-Ch-Ua-Platform: ""

Accept: image/avif, image/webp, image/apng, image/svg+xml, image/\*, \*/\*;q=0.8

Sec-Fetch-Site: same-origin

Inspector pane shows: Request attributes (2), Request cookies (5), Request headers (20), Response headers (20).