

**Finding Name:** Cross Origin Resource sharing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Team** | **Role** | **Project** | **Quality Assurance** | **Is this a re-tested Finding?** |
| Vishnu Madhusoodanan Nair | PT | Junior Team Member | Ontrack | Nabiha Masood and Oliver Power | Yes |

|  |
| --- |
| **Was this Finding Successful?** |
| Yes |

**Finding Description**

An HTML5 cross-origin resource sharing (CORS) policy controls whether and how content running on other domains can perform two-way interaction with the domain that publishes the policy. Because of its fine-grained nature, the policy can apply access controls per-request depending on the URL and other request characteristics.   
If the policy permits access to a different domain, users of the application may be attacked by that domain. When a user logs into the application and visits a domain that has been approved by the policy, any malicious content that is operating on that domain may be able to retrieve content from the application and occasionally perform actions within the logged-in user's security context.

An attacker may be able to take advantage of security flaws in an approved domain, even if it isn't explicitly malevolent, to break the trust relationship and target the application that grants access.

The application permits any origin to be added to the "Access-Control-Allow-Origin:" header and to be trusted. This invalidates any policy enforced by CORS for this application. An attacker may trick a user into clicking on a link that contains malicious JavaScript that sends a request in an attempt to delete, modify, or obtain personal data.

Below are the pages noticed with “Access-Control-Allow-Origin: \* ”.

|  |  |
| --- | --- |
| Pages/Route | /api/auth |
| /api/auth/signout\_url |
| /api/campuses/ |
| /api/projects/ |
| /api/settings |
| /api/teaching\_periods/ |
| /api/unit\_roles/ |

Request:

GET /api/unit\_roles/ HTTP/1.1

Host: localhost:3000

User-Agent: Mozilla/5.0 (X11; Linux x86\_64; rv:109.0) Gecko/20100101 Firefox/115.0

Accept: application/json, text/plain, \*/\*

Accept-Language: en-US,en;q=0.5

Accept-Encoding: gzip, deflate

Auth-Token: DxKH\_sKcb7zLWMmrExBo

Username: student\_1

Origin: http://google.com

Connection: close

Referer: http://localhost:4200/

Sec-Fetch-Dest: empty

Sec-Fetch-Mode: cors

Sec-Fetch-Site: same-site

If-None-Match: W/"4f53cda18c2baa0c0354bb5f9a3ecbe5"

**Risk Rating**  
Impact: Major  
Likelihood: High

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Impact values** | | | | |
| **Very Minor** | **Minor** | **Significant** | **Major** | **Severe** |
| Risk that holds little to no impact. Will not cause damage and regular activity can continue. | Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity. | Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally. | Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally. | Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Likelihood** | | | | |
| **Rare** | **Unlikely** | **Moderate** | **High** | **Certain** |
| Event may occur and/or if it did, it happens in specific circumstances. | Event could occur occasionally and/or could happen (at some point) | Event may occur and/or happens. | Event occurs at times and/or probably happens a lot. | Event is occurring now and/or happens frequently. |

**Business Impact**

Attacker may be able to pass vulnerable URL link to the payload and hack the application easily.

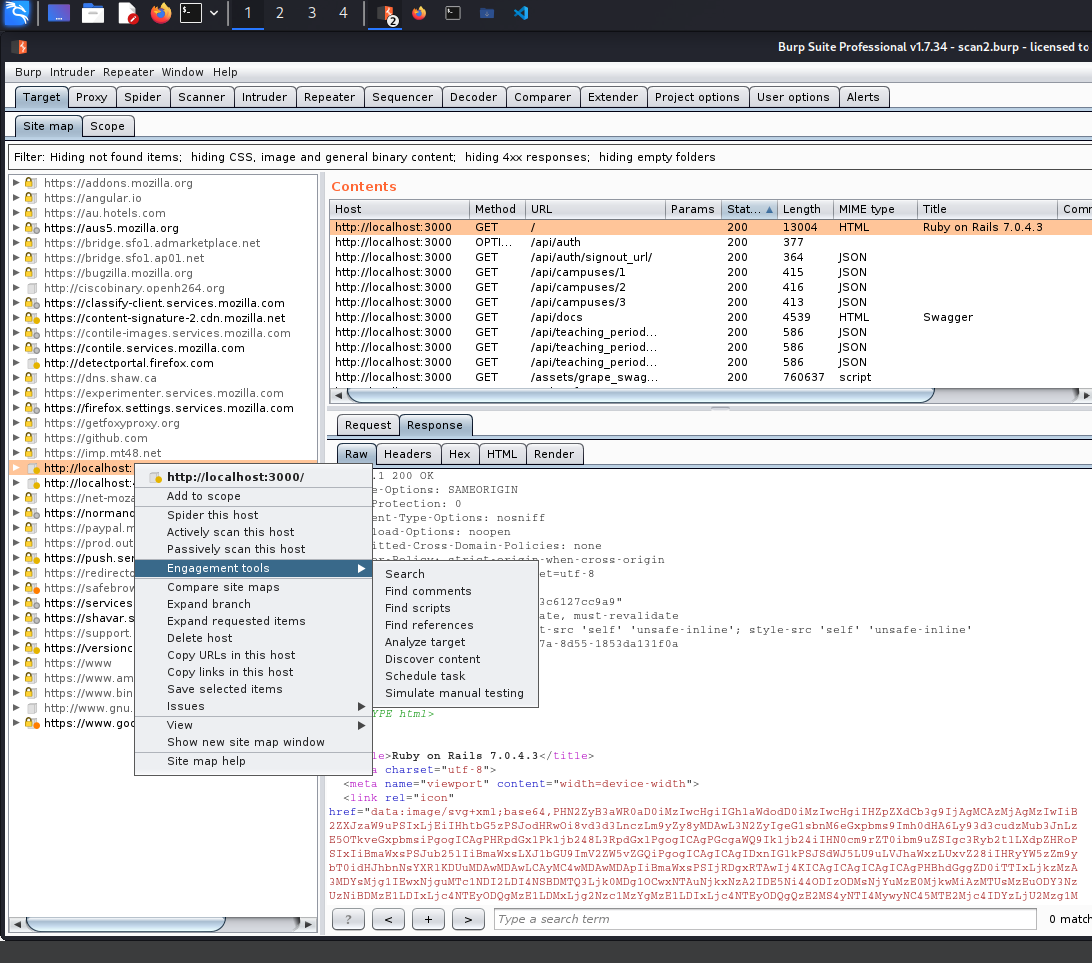
**Affected Assets**

Web Application.

**Evidence**

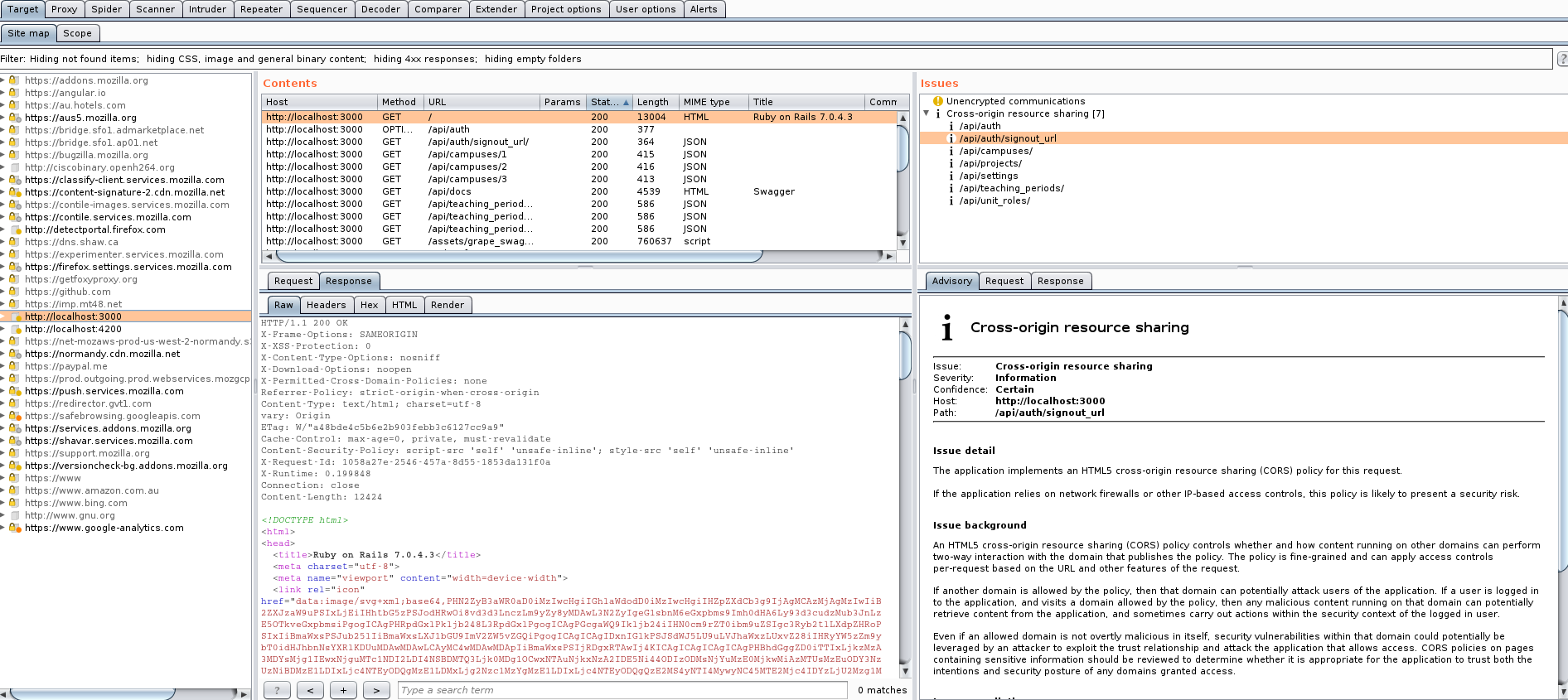
**Step 1.**

Login as student and create a burp scan. Go to target tab. Right click on endpoint and go to engagement tools 🡪 Discover content



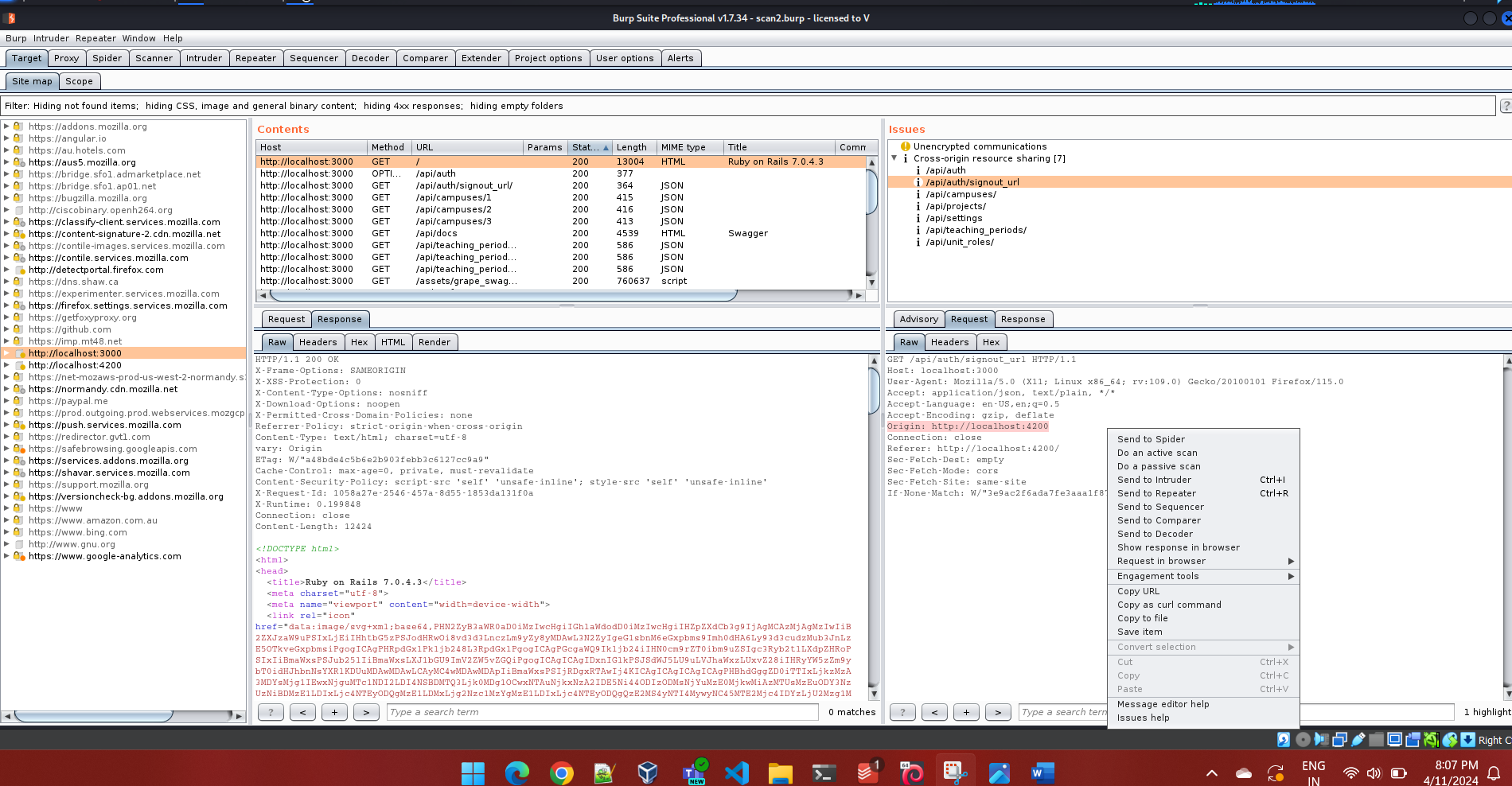
**Step 2.**

Let the scan run for a while. After some time, we can see the issues pop up on the issues tab.



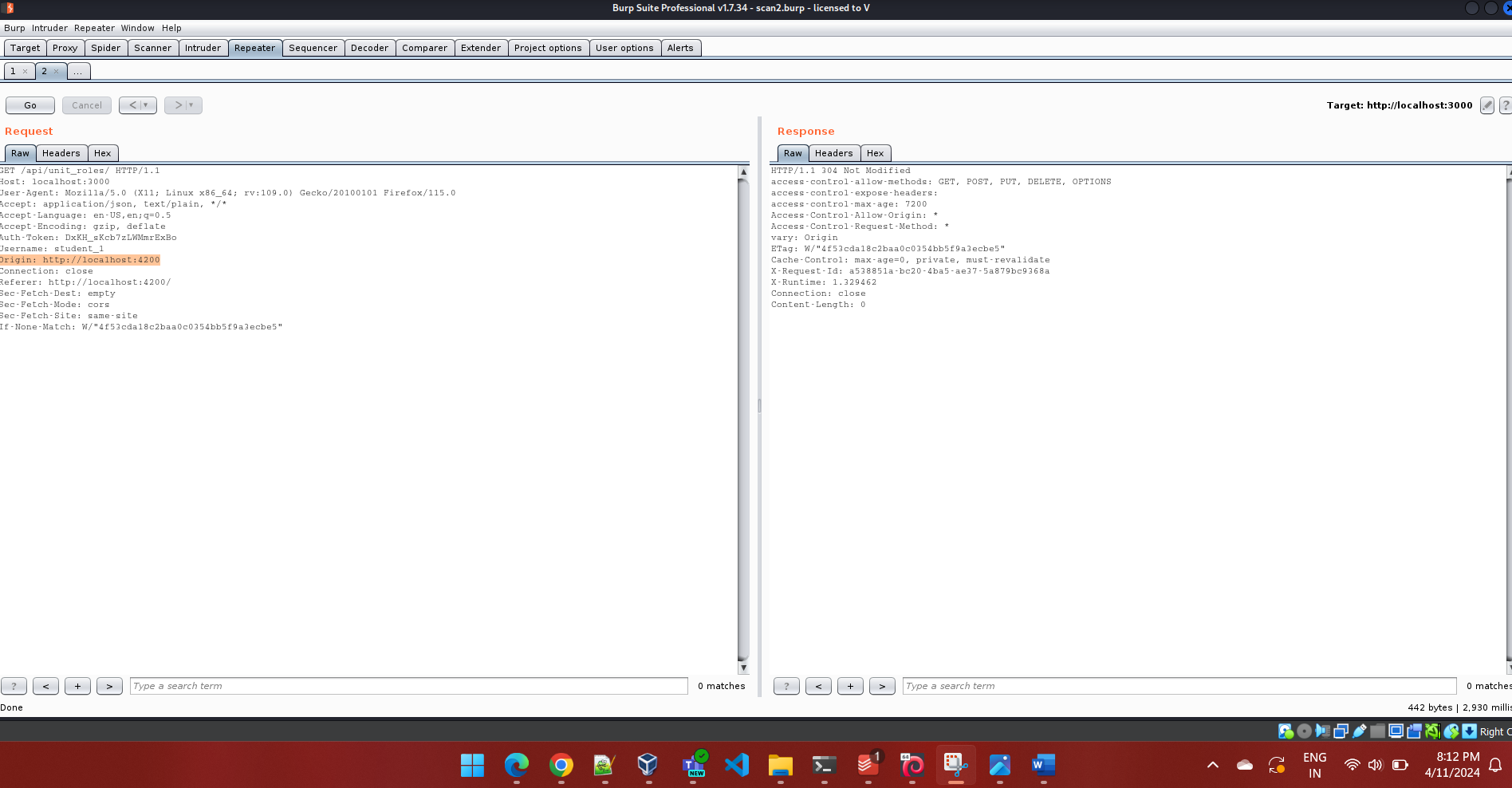
**Step 3.**

Click on any of the Cross origin request sharing issue. Go to Request tab. Right click and click on option sent to repeater



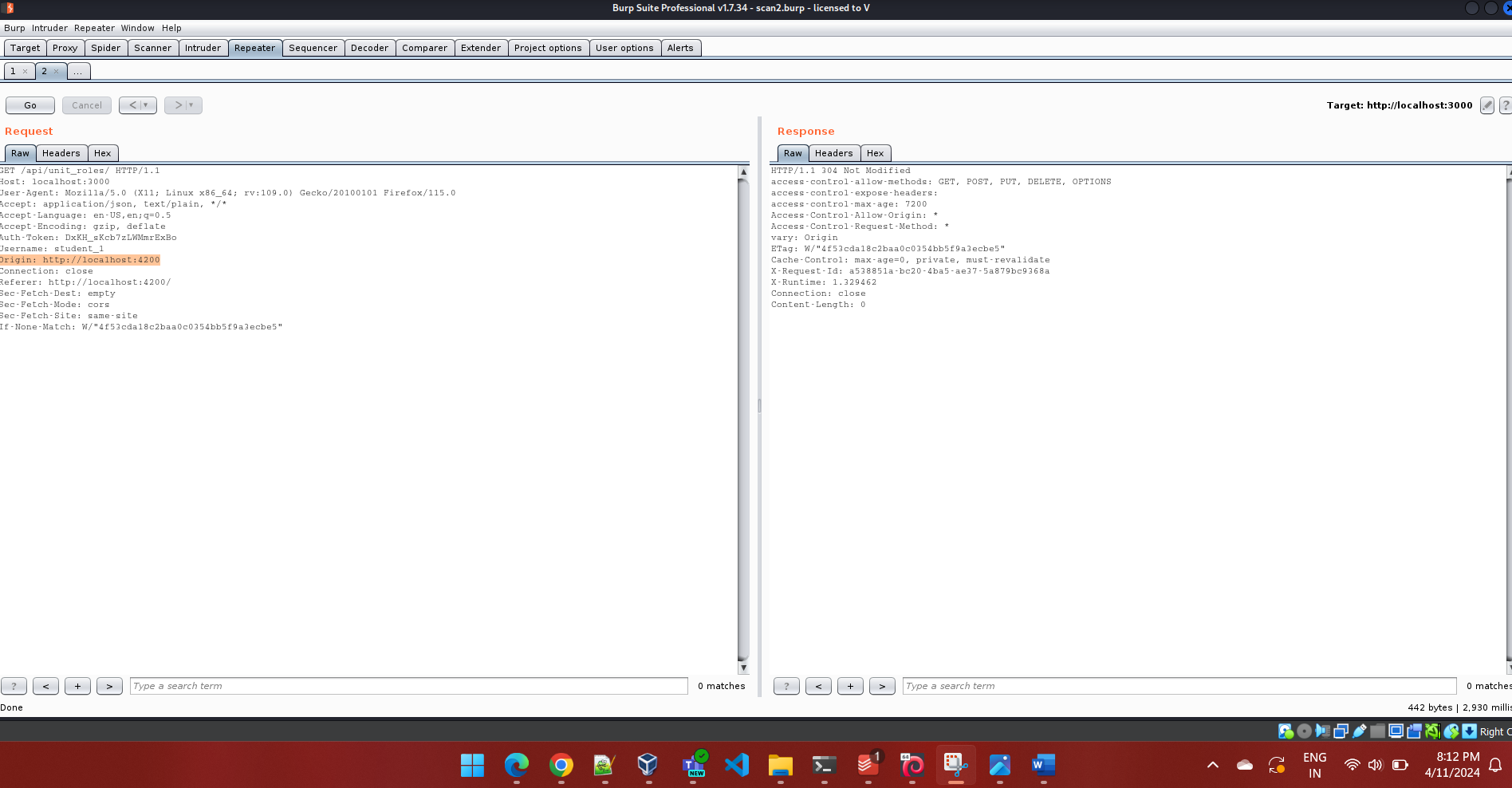
**Step 4.**

Hit Go on the payload that has come on the repeater. We will see the response.



**Step 5**.

Now change the payload origin param to <http://google.com> or any other website of our chpoice and hit sent. Now also we are able to see response as the Access-Control-Allow-Origin is set to \*.



**Remediation Advice**

Update policy to permit only trusted domains for cross-domain access rather than providing access to all(\*).

**References**

Tools Used : Burpsuite

Reference

1) <https://www.owasp.org/index.php/HTML5_Security_Cheat_Sheet#Cross_Origin_Resource_Sharing>

**Contact Details**

Name : Vishnu Madhusoodanan Nair

Email : s223250182@deakin.edu.au

**Pentest Leader Feedback.**