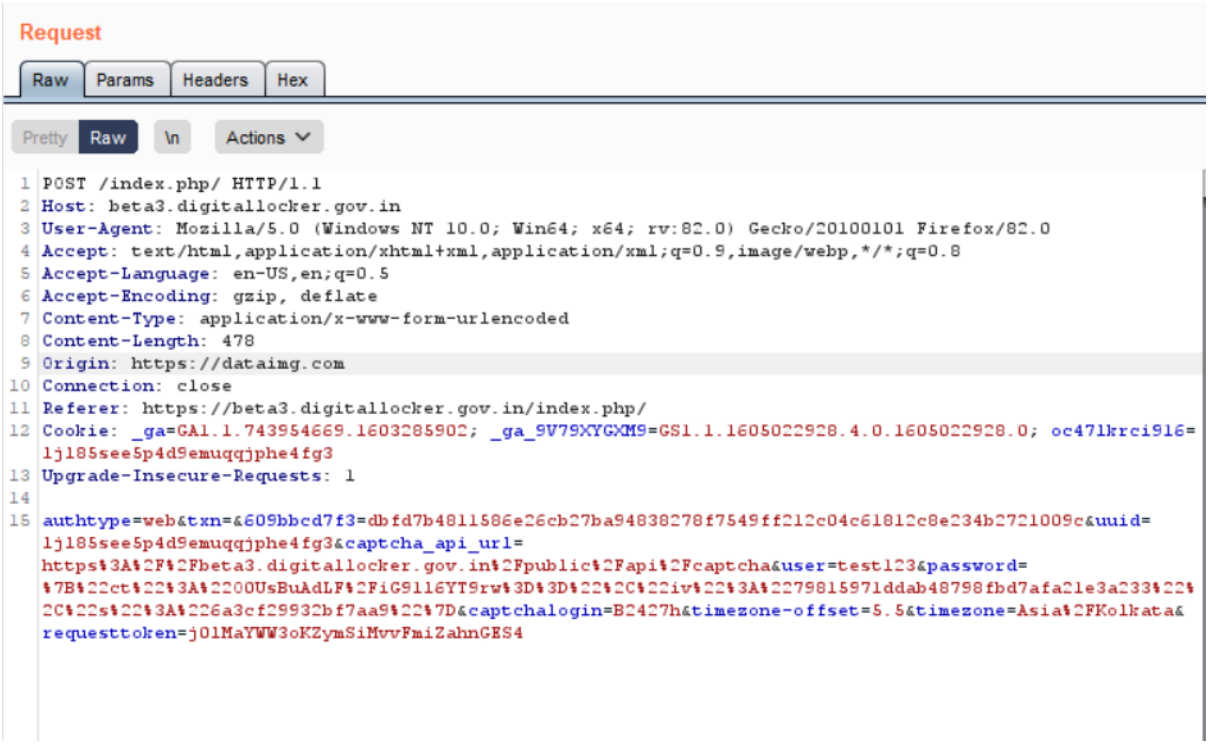
CORS (Cross-origin resource sharing)

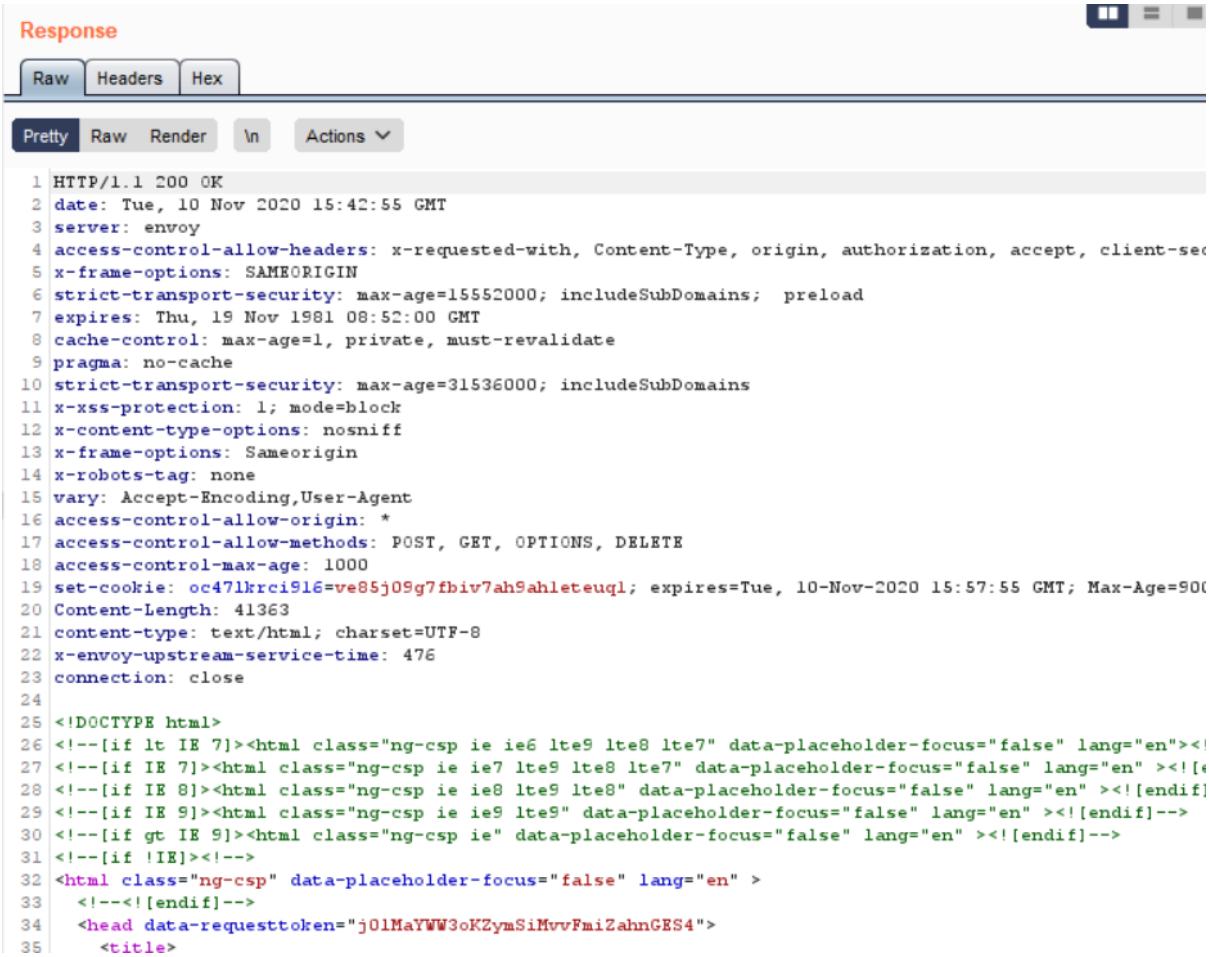
Summary: Cross-origin resource sharing (CORS) is a browser mechanism which enables controlled access to resources located outside of a given domain. However, it also provides potential for cross-domain based attacks, if a website's CORS policy is poorly configured and implemented. CORS can be exploited to trust any arbitrary domain attacker controlled domain name and send the data to it. Attackers can make an exploit and ask the domain to send data of the victim to the attacker domain.

Severity: Low

Request



Response



As you can see when we run the above request in curl we can see these header results in the response.

Access-Control-Allow-Origin: *

Complexity: Easy

Proof of Concept: Attached in the Video

Video is attached with mail

Impact: It is security misconfiguration no further exploit is possible on victim is possible

Affected Host:

https://beta3.digitallocker.gov.in/index.php/

Recommendations:

a. All the REST APIs should be authenticated and the domain should not trust any other domains. Allow only selected, trusted domains in the Access-Control-Allow-Origin header.

b. To mitigate the risk of CORS, we always recommend whitelisting your Access-Control-Allow-Origin instead of wildcarding. Using a wildcard prefix such as *.yoursite.com makes it more difficult for the attackers given they would need to find a vulnerability (such as cross-site scripting or cross-site request forgery) to issue the cross-origin request.

References:

1. https://owasp.org/www-community/attacks/CORS_OriginHeaderScrutiny
2. https://www.geekboy.ninja/blog/exploiting-misconfigured-cors-cross-origin-resource-sharing/
3. https://en.wikipedia.org/wiki/Cross-origin_resource_sharing
4. https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS