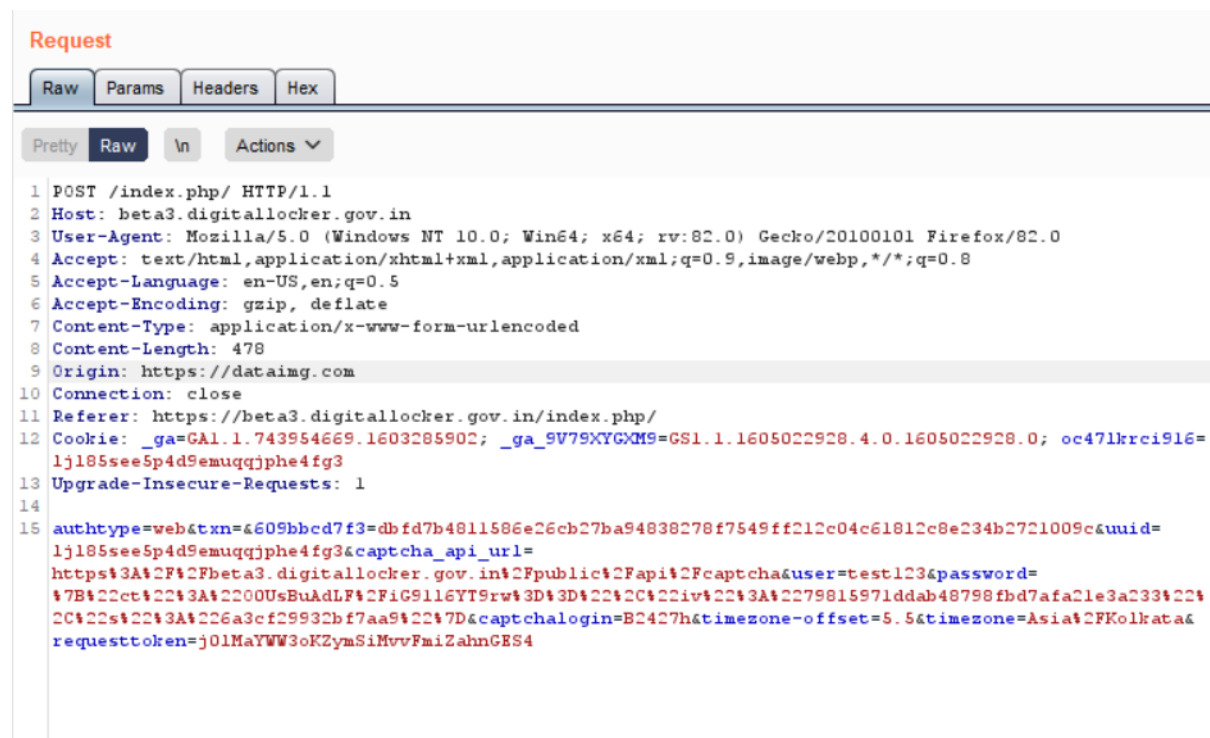


CORS (Cross-origin resource sharing)

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



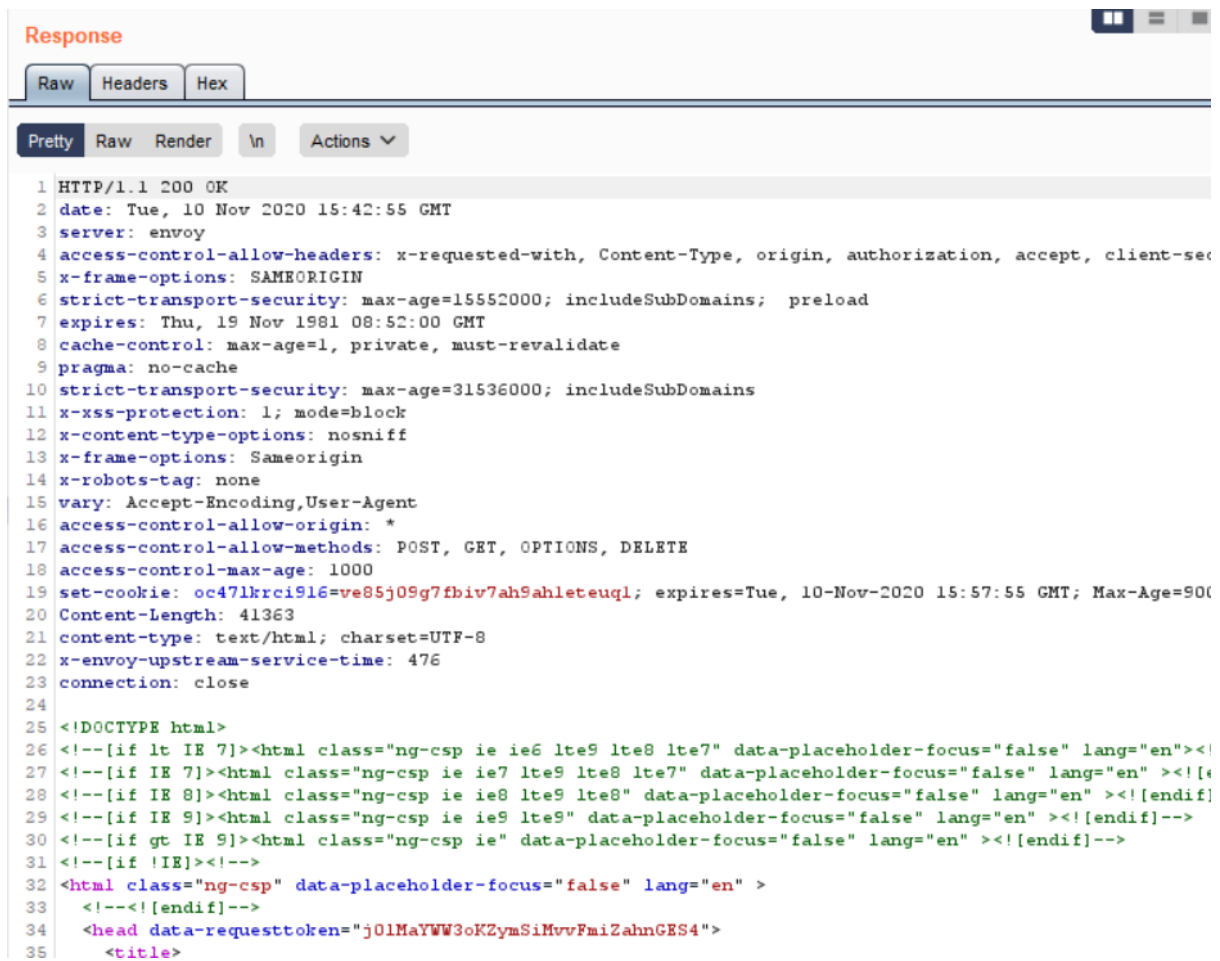
Request

Raw Params Headers Hex

Pretty Raw \n Actions

```
1 POST /index.php/ HTTP/1.1
2 Host: beta3.digitallocker.gov.in
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:82.0) Gecko/20100101 Firefox/82.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 478
9 Origin: https://dataimg.com
10 Connection: close
11 Referer: https://beta3.digitallocker.gov.in/index.php/
12 Cookie: _ga=GA1.1.743954669.1603285902; _ga_9V79XYGXM9=GS1.1.1605022928.4.0.1605022928.0; oc47lkrcl916=
13 Upgrade-Insecure-Requests: 1
14
15 authtype=web&tzn=&609bbcd7f3=dbfd7b4811586e26cb27ba94838278f7549ff212c04c61812c8e234b2721009c&uuid=
16 1j185see5p4d9emuqqjphe4fg3&captcha_api_url=
17 https%3A%2F%2Fbeta3.digitallocker.gov.in%2Fpublic%2Fapi%2Fcaptcha&user=test123&password=
18 %7B%22ct%22%3A%2200UsBuAdLF%2FiG9116YT9rw%3D%3D%22%2C%22iv%22%3A%2279815971ddab48798fbd7afa21e3a233%22%
19 2C%22s%22%3A%226a3cf29932bf7aa9%22%7D&captcha_login=B2427h&timezone-offset=5.5&timezone=Asia%2FKolkata&
20 requesttoken=j01MaYWW3oKZymSiMvvFmiZahnGES4
```

Response



```
1 HTTP/1.1 200 OK
2 date: Tue, 10 Nov 2020 15:42:55 GMT
3 server: envoy
4 access-control-allow-headers: x-requested-with, Content-Type, origin, authorization, accept, client-se
5 x-frame-options: SAMEORIGIN
6 strict-transport-security: max-age=15552000; includeSubDomains; preload
7 expires: Thu, 19 Nov 1981 08:52:00 GMT
8 cache-control: max-age=1, private, must-revalidate
9 pragma: no-cache
10 strict-transport-security: max-age=31536000; includeSubDomains
11 x-xss-protection: 1; mode=block
12 x-content-type-options: nosniff
13 x-frame-options: Sameorigin
14 x-robots-tag: none
15 vary: Accept-Encoding,User-Agent
16 access-control-allow-origin: *
17 access-control-allow-methods: POST, GET, OPTIONS, DELETE
18 access-control-max-age: 1000
19 set-cookie: oc47lkrCi9l6=ve85j09g7fbiv7ah9ahleteuql; expires=Tue, 10-Nov-2020 15:57:55 GMT; Max-Age=90(
20 Content-Length: 41363
21 content-type: text/html; charset=UTF-8
22 x-envoy-upstream-service-time: 476
23 connection: close
24
25 <!DOCTYPE html>
26 <!--[if lt IE 7]><html class="ng-csp ie ie6 lte9 lte8 lte7" data-placeholder-focus="false" lang="en"><
27 <!--[if IE 7]><html class="ng-csp ie ie7 lte9 lte8 lte7" data-placeholder-focus="false" lang="en" ><![
28 <!--[if IE 8]><html class="ng-csp ie ie8 lte9 lte8" data-placeholder-focus="false" lang="en" ><![endif]
29 <!--[if IE 9]><html class="ng-csp ie ie9 lte9" data-placeholder-focus="false" lang="en" ><![endif]-->
30 <!--[if gt IE 9]><html class="ng-csp ie" data-placeholder-focus="false" lang="en" ><![endif]-->
31 <!--[if !IE]><!-->
32 <html class="ng-csp" data-placeholder-focus="false" lang="en" >
33 <!--<![endif]-->
34 <head data-requesttoken="j0lMaYWW3oKZymSiMvvFmiZahnGES4">
35 <title>
```

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:

- 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>