Title: - Security Misconfiguration (Cross origin Resource Sharing)

Description: - The Access-Control-Allow-Origin header is included in the response from one website to a request originating from another website, and identifies the permitted origin of the request. A web browser compares the Access-Control-Allow-Origin with the requesting website's origin and permits access to the response if they match.

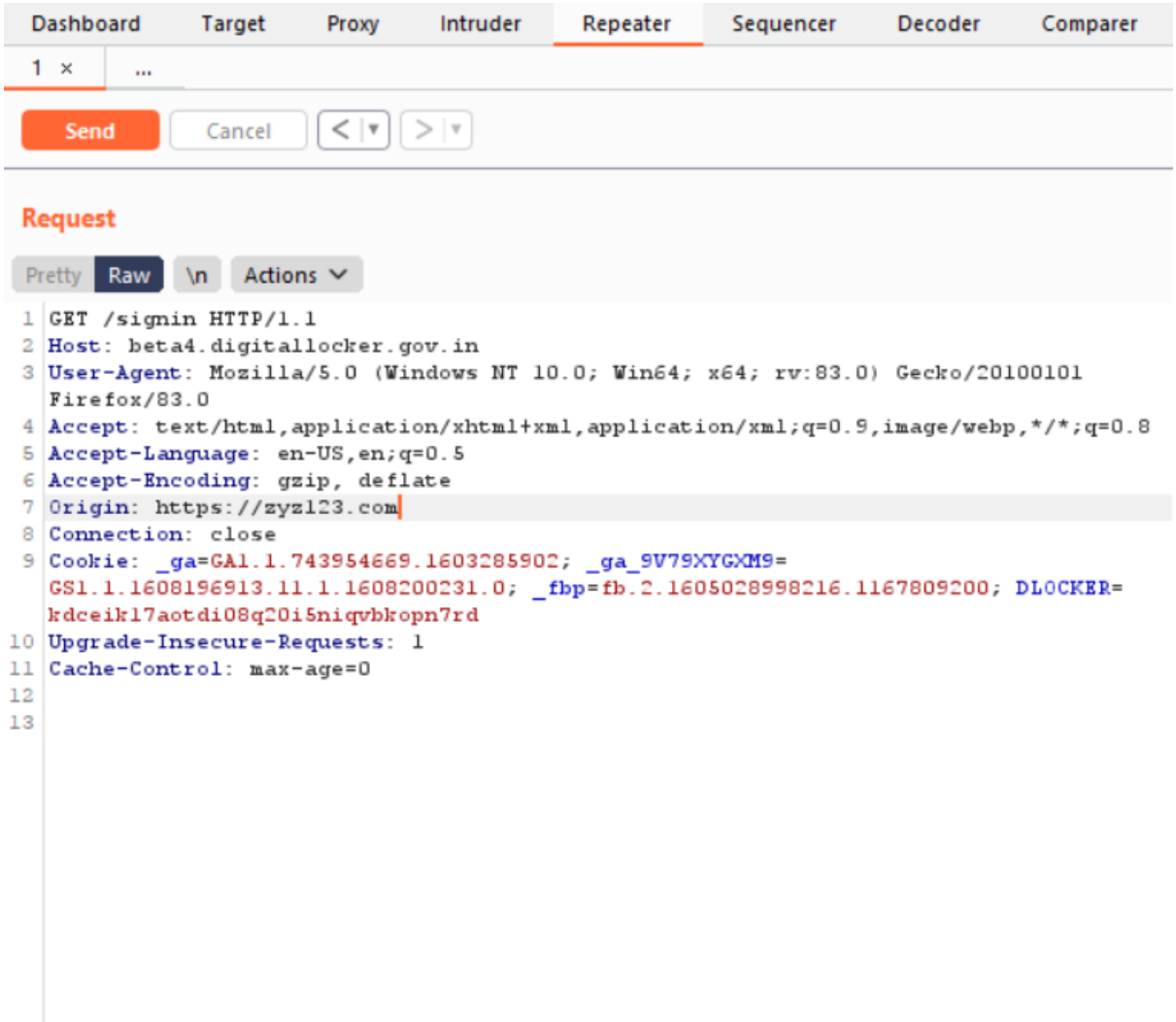
The header Access-Control-Allow-Origin supports wildcards. For example:

Access-Control-Allow-Origin: *

Note: - wildcards cannot be used within any other value. For example, the following header is not valid:

Access-Control-Allow-Origin: https://*.normal-website.com

Request



Response

Response

Pretty Raw Render \n Actions

```
1  HTTP/1.1 200 OK
2  server: envoy
3  date: Thu, 17 Dec 2020 10:17:46 GMT
4  content-type: text/html; charset=UTF-8
5  vary: Accept-Encoding
6  expires: Thu, 19 Nov 1981 08:52:00 GMT
7  cache-control: no-store, no-cache, must-revalidate
8  pragma: no-cache
9  x-frame-options: SAMEORIGIN
10 x-xss-protection: 1; mode=block
11 x-content-type-options: nosniff
12 strict-transport-security: max-age=31536000; env=HTTPS;
13 access-control-allow-origin: *
14 access-control-allow-methods: POST, GET, OPTIONS, DELETE, PUT
15 access-control-allow-headers: x-requested-with, Content-Type, origin, authorizat.
16 x-envoy-upstream-service-time: 68
17 connection: close
18 Content-Length: 47997
19
20 <!DOCTYPE html>
21 <html xml:lang="en" lang="en">
22
23   <head>
24     <meta charset="utf-8">
25     <meta name="viewport" content="width=device-width, initial-scale=1.0, minimu
26     <title>
       DigiLocker
       </title>
27     <!-- global style css -->
```

Impact: -

Fortunately, from a security perspective, the use of the wildcard is restricted in the specification as you cannot combine the wildcard with the cross-origin transfer of credentials (authentication, cookies or client-side certificates). Consequently, a cross-domain server response of the form:

Access-Control-Allow-Origin: *

It is not permitted as this would be dangerously insecure, exposing any authenticated content on the target site to everyone.

Given these constraints, some web servers dynamically create Access-Control-Allow-Origin headers based upon the client-specified origin. This is a workaround for CORS constraints that is not secure.