[Skip to main content](https://lms.alnafi.com/xblock/block-v1:alnafi+DCCS102+2025_DCCS+type@vertical+block@46274f0c97914e30933e7a0046042d43?exam_access=&recheck_access=1&show_bookmark=0&show_title=0&view=student_view#main)

**Testing Cross-Origin Resource Sharing**

**Summary**

[Cross origin resource sharing](https://en.wikipedia.org/wiki/Cross-origin_resource_sharing) (CORS) is a mechanism that enables a web browser to perform cross-domain requests using the XMLHttpRequest L2 API in a controlled manner. In the past, the XMLHttpRequest L1 API only allowed requests to be sent within the same origin as it was restricted by the same origin policy.

Cross-origin requests have an origin header that identifies the domain initiating the request and is always sent to the server. CORS defines the protocol to use between a web browser and a server to determine whether a cross-origin request is allowed. HTTP [headers](https://en.wikipedia.org/wiki/Cross-origin_resource_sharing#Headers) are used to accomplish this.

The [W3C CORS specification](https://www.w3.org/TR/cors/) mandates that for non simple requests, such as requests other than GET or POST or requests that uses credentials, a pre-flight OPTIONS request must be sent in advance to check if the type of request will have a bad impact on the data. The pre-flight request checks the methods and headers allowed by the server, and if credentials are permitted. Based on the result of the OPTIONS request, the browser decides whether the request is allowed or not.

**Origin & Access-Control-Allow-Origin**

The origin header is always sent by the browser in a CORS request and indicates the origin of the request. The origin header can not be changed from JavaScript however relying on this header for Access Control checks is not a good idea as it may be spoofed outside the browser, so you still need to check that application-level protocols are used to protect sensitive data.

Access-Control-Allow-Origin is a response header used by a server to indicate which domains are allowed to read the response. Based on the CORS W3 Specification it is up to the client to determine and enforce the restriction of whether the client has access to the response data based on this header.

From a penetration testing perspective you should look for insecure configurations as for example using a \* wildcard as value of the Access-Control-Allow-Origin header that means all domains are allowed. Other insecure example is when the server returns back the origin header without any additional checks, what can lead to access of sensitive data. Note that this configuration is very insecure, and is not acceptable in general terms, except in the case of a public API that is intended to be accessible by everyone.

**Access-Control-Request-Method & Access-Control-Allow-Method**

The Access-Control-Request-Method header is used when a browser performs a preflight OPTIONS request and let the client indicate the request method of the final request. On the other hand, the Access-Control-Allow-Method is a response header used by the server to describe the methods the clients are allowed to use.

**Access-Control-Request-Headers & Access-Control-Allow-Headers**

These two headers are used between the browser and the server to determine which headers can be used to perform a cross-origin request.