URL:

http://vhldvdofe009:8080/message/messageSummary/

Method:

OPTIONS

Vulnerable Parameter:

Attack Payload:

http://vhldvdofe009:8080/message/messageSummary/

**REQUEST**

OPTIONS http://vhldvdofe009:8080/message/messageSummary/ HTTP/1.1

Accept-Encoding: gzip,deflate

Content-Type: application/json

Host: vhldvdofe009:8080

User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

Accept: \*/\*

Pragma: no-cache

Access-Control-Request-Method: POST

Access-Control-Request-Headers: X-Pingsession

Origin: null

Referer: http://vhldvdofe009:8080/message/messageSummary/

Connection: Keep-Alive

Content-Length: 290

X-WIPP: AscVersion=18.10.127.0

X-Scan-Memo: Category="Audit.Attack"; SID="8BF8FA6A5FBF5F886425F20F1B8E1288"; PSID="021CEBF258158E082CAC13319303EB06"; SessionType="AuditAttack"; CrawlType="None"; AttackType="Other"; OriginatingEngineID="822a8e1c-b895-4666-a9d2-026b0a4716c9"; AttackSequence="0"; AttackParamDesc=""; AttackParamIndex="0"; AttackParamSubIndex="0"; CheckId="11281"; Engine="Html5+Cross+Origin+Options+Request"; SmartMode="NonServerSpecificOnly"; ThreadId="134"; ThreadType="AuditorStateRequestor";

X-RequestManager-Memo: RequestorThreadIndex="1"; sc="1"; ID="9d93fb3a-ed81-4597-a45d-61540bfd8b8a";

X-Request-Memo: ID="52440a43-7e11-48e7-86f7-2eccb7eeda70"; sc="1"; ThreadId="134";

Cookie: CustomCookie=WebInspect0

{"monitoredSystem":{"name":"4DL"},"userId":{"syscode\_id":"1","process\_id":"1","condition\_id":"1"},"fromDate":{"year":"2019","month":"04","day":"09","hour":"15","min":"00"},"toDate":{"year":"2019","month":"04","day":"09","hour":"18","min":"00"},"dataConditionsIsOr":true,"dataConditions":[]}

**RESPONSE**

HTTP/1.1 200

Vary: Origin

Vary: Access-Control-Request-Method

Vary: Access-Control-Request-Headers

Access-Control-Allow-Origin: null

Access-Control-Allow-Methods: GET,HEAD,POST

Access-Control-Allow-Headers: X-Pingsession

Access-Control-Allow-Credentials: true

Access-Control-Max-Age: 1800

Allow: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH

Date: Thu, 09 Apr 2020 16:22:06 GMT

Content-Length: 5

**OVERVIEW**

WebInspect has detected the target application supports “**Origin: null**” for CORS requests, making it vulnerable to CORS attacks.  
  
Cross-Origin Resource Sharing, commonly referred to as CORS, is a technology that allows a domain to define a policy for its resources to be accessed by a web page hosted on a different domain using cross domain XML HTTP Requests (XHR). Historically, the browser restricts cross domain XHR requests to abide by the same origin policy. At its basic form, the same origin policy sets the script execution scope to the resources available on the current domain and prohibits any communication to domains outside this scope. Therefore, execution and incorporation of remote methods and functions hosted on domains outside of the current domain are effectively prohibited. While CORS is supported on all major browsers, it also requires that the domain correctly defines the CORS policy in order to have its resources shared with another domain. These restrictions are managed by access policies typically included in specialized response headers, such as:

* Access-Control-Allow-Origin

* Access-Control-Allow-Headers

* Access-Control-Allow-Methods

In this instance, the **Access-Control-Allow-Origin** header is set to "**null**" as seen in the pre-flight response. This makes the target application vulnerable to CORS attacks. Sandboxed documents and various URI scheme such as data: or file: have their 'Origin' defined as ‘null’. Thus, any malicious user can easily obtain "**Origin: null**" can be easily obtained by any malicious user by simply enclosing the malicious CORS request from within an *iframe* and using the *sandbox* attribute.  
  
Example:  
  
<iframe sandbox="allow-scripts allow-forms" src="data:text/html, <!DOCTYPE html> <script>var req = new XMLHttpRequest();</script>" />

**RECOMMENDATION**

You should disallow access to CORS requests with the “**Origin: null**” header especially in the case of credentialed requests.