WWBN / AVideo Public

Code Issues 13 Pull requests Actions Projects Wiki Security a

Commit caf705f

Daniel Neto committed last week · 11 / 11

fix: Enhance CORS security by validating origins for credentialed requests

GHSA-ccq9-r5cw-5hwq

master

1 parent 8d8fc0c commit caf705f

1 file changed +25 -7 lines changed

Top

Filter files…

objects

functions.php

1 file changed +25 -7 lines changed

Search within code

objects/functions.php

```
@@ -2764,18 +2764,36 @@ function allowOrigin($allowAll = false)
2764 2764
2765 2765         // Public resources (e.g. VAST/VMAP ad XML) should be readable by any
2766 2766         // origin.  Pass $allowAll = true for permissive CORS.
2767      -     // When the browser sends a credentialed request (credentials:'include', e.g.
2768      -     // the IMA SDK), it rejects the wildcard '*' – the spec requires echoing the
2769      -     // exact origin in that case.  We reflect whatever origin is in the request
2770      -     // and add Allow-Credentials:true so credentialed fetches also succeed.
2771      -     // These endpoints return public ad XML and carry no session-sensitive data,
```

```
2772      -     // so reflecting any origin is safe here.
      2767  +     // SECURITY: even in $allowAll mode we must NOT reflect an arbitrary third-party
      2768  +     // Origin together with Access-Control-Allow-Credentials:true — that lets any
      2769  +     // attacker page make credentialed cross-origin requests and read session-
      2770  +     // authenticated API responses (user PII, stream keys, admin flags).
      2771  +     // We therefore validate the origin the same way as the $allowAll=false path:
      2772  +     // - Same-origin requests get reflected + credentials (logged-in browser calls)
      2773  +     // - All other origins get wildcard without credentials (public/ad-tech reads)
      2774  +     // Mobile apps use APISecret token auth, not session cookies, so they are
      2775  +     // unaffected by removing Allow-Credentials for third-party origins.
2773  2776        if ($allowAll) {
2774  2777            $requestOrigin = $_SERVER['HTTP_ORIGIN'] ?? '';
2775      -         if (!empty($requestOrigin)) {
      2778  +
      2779  +         $siteOriginForAllowAll = '';
      2780  +         if (!empty($global['webSiteRootURL'])) {
      2781  +             $parsedForAllowAll = parse_url($global['webSiteRootURL']);
      2782  +             if (!empty($parsedForAllowAll['scheme']) && !empty($parsedForAllowAll['host'])) {
      2783  +                 $siteOriginForAllowAll = $parsedForAllowAll['scheme'] . '://' . $parsedForAllowAll['host'];
      2784  +                 if (!empty($parsedForAllowAll['port'])) {
      2785  +                     $siteOriginForAllowAll .= ':' . $parsedForAllowAll['port'];
      2786  +                 }
      2787  +             }
      2788  +         }
      2789  +
      2790  +         if (!empty($requestOrigin) && !empty($siteOriginForAllowAll) && $requestOrigin === $siteOriginForAllowAll) {
      2791  +             // Verified same-origin request — reflect with credentials
2776  2792                header('Access-Control-Allow-Origin: ' . $requestOrigin);
2777  2793                header('Access-Control-Allow-Credentials: true');
2778  2794            } else {
```

```
        2795 +             // Third-party or no origin: allow non-credentialed reads only.
        2796 +             // This covers IMA/ad-tech fetches (VAST/VMAP) which never send
               credentials.
2779    2797               header('Access-Control-Allow-Origin: *');
2780    2798           }
2781    2799           header('Access-Control-Allow-Private-Network: true');
```

Comments 0

Please sign in to comment.