Cross-origin resource sharing Vulnerability

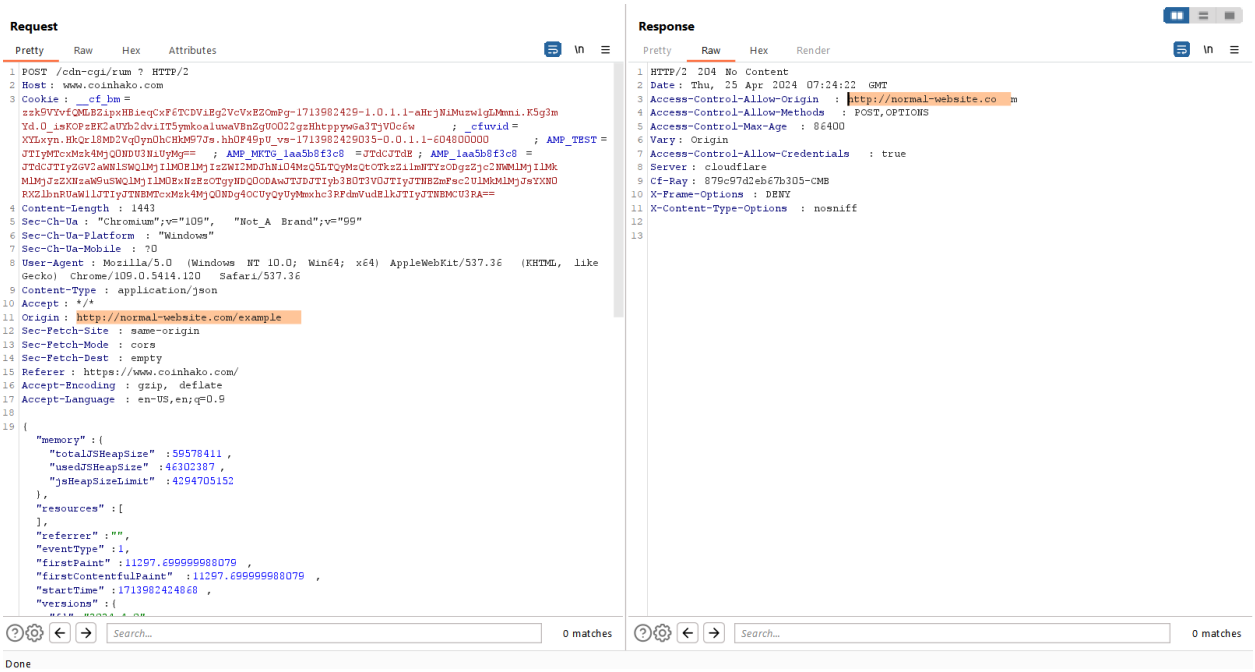
Vulnerability description:

The application's cross-origin resource sharing (CORS) policy permits access from any domain, including the requested origin http://normal-website.com:8080/example/.

Affected components:

cross-origin resource sharing (CORS) policy

Proof of concept



Proposed mitigation or fix

- Whitelisting Origins: Implement a whitelist of trusted origins using the Access-Control-Allow-Origin header. Specify only the domains that are explicitly permitted to access resources. Avoid setting the header to NULL, as it opens the door to potential exploitation by malicious actors.
- Method Validation: Use the Access-Control-Allow-Methods header to specify the HTTP methods allowed for approved origins. Different domains may require different levels of access (e.g., read-only vs. read-write). By explicitly defining permitted methods, you can mitigate the risk of unauthorized actions.
- Continuous Monitoring: Regularly review the CORS headers in your application's responses to ensure they align with your security policies. Validate the values of these headers to detect any misconfigurations or vulnerabilities. Consider leveraging open-source scanners to automate this process and identify potential security gaps.

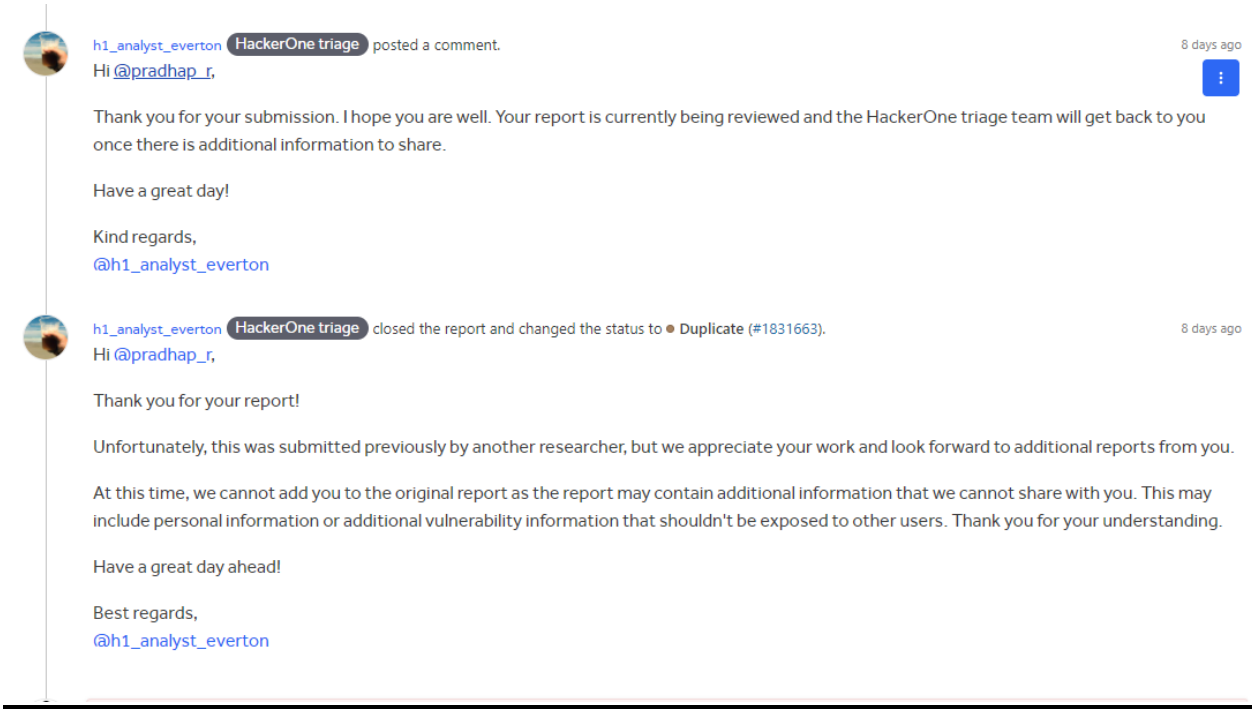
Impact:

- Access Sensitive Data: Retrieve confidential information from other origins.
- Execute CSRF Attacks: Perform unauthorized actions on behalf of authenticated users.
- Leak Information: Disclose sensitive details about the target system.
- Hijack Sessions: Steal session tokens to impersonate legitimate users.
- Exploit XSS: Facilitate data theft by exfiltrating stolen data to remote domains.
- Manipulate Data: Modify or delete critical resources on vulnerable servers.
-

Conclusion:

Securing web applications against CORS-based attacks requires careful configuration and continuous monitoring. By implementing a whitelist-based approach, validating permitted methods, and regularly reviewing CORS headers, organizations can mitigate the risk of unauthorized access and protect sensitive data from exploitation.

Response form HackerOne



h1_analyst_everton HackerOne triage posted a comment. 8 days ago

Hi @pradhap_r,

Thank you for your submission. I hope you are well. Your report is currently being reviewed and the HackerOne triage team will get back to you once there is additional information to share.

Have a great day!

Kind regards,
@h1_analyst_everton

h1_analyst_everton HackerOne triage closed the report and changed the status to ● Duplicate (#1831663). 8 days ago

Hi @pradhap_r,

Thank you for your report!

Unfortunately, this was submitted previously by another researcher, but we appreciate your work and look forward to additional reports from you.

At this time, we cannot add you to the original report as the report may contain additional information that we cannot share with you. This may include personal information or additional vulnerability information that shouldn't be exposed to other users. Thank you for your understanding.

Have a great day ahead!

Best regards,
@h1_analyst_everton