Custom Request Header

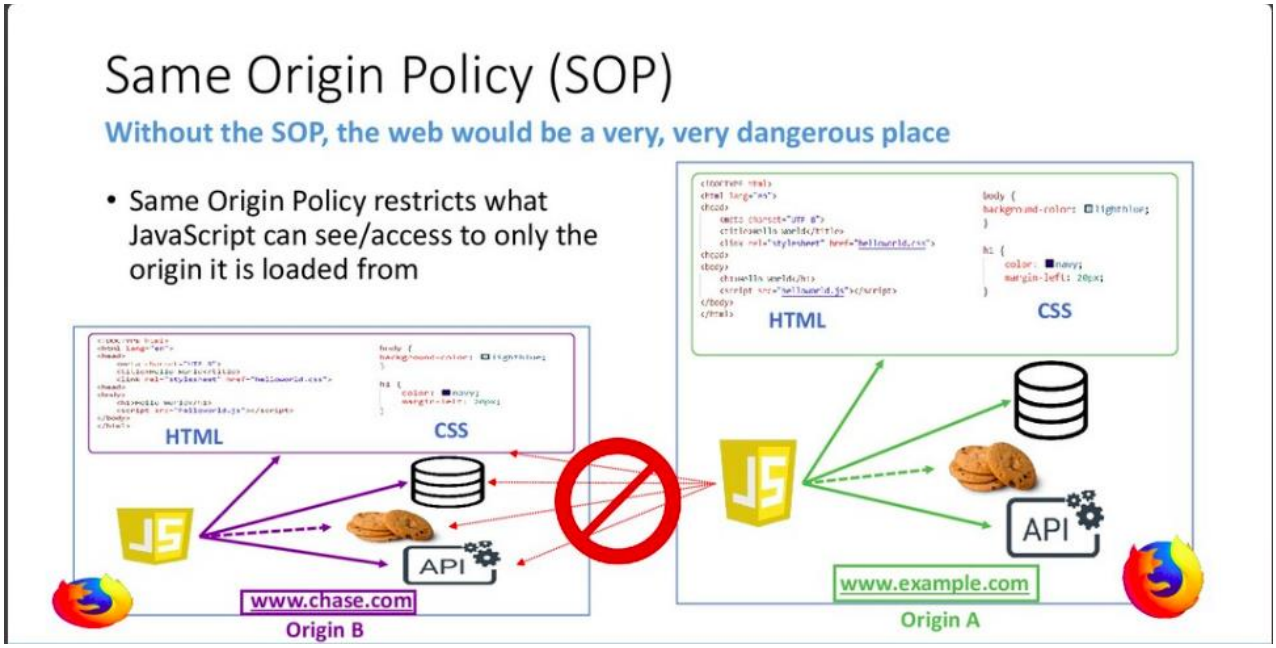
I. What is Custom Request Header?

Custom Request Header is a solution to protect against CSRF attacks. This approach has the double advantage of usually requiring no UI changes and not introducing any server side state. You can always add your own **custom header** and value if that is preferred.

II. How does it works ?

JavaScript can send custom request headers along with the request to the webserver. To prevent malicious requests, you can simply check the presence and the value of this header on all your AJAX server-side endpoints.

A widely used custom header which can be used is: **X-Requested-With: XMLHttpRequest**. This defense relies on the same-origin policy restriction which prevents browsers to make cross origin requests and the restriction that only JavaScript can add custom headers to a request.



Example: Using XMLhttprequest object

```
<body>
    <p>This form is submitted with JavaScript and utilzes the
        "custom header" CSRF protection mechanism</p>
    <form name="transfer" method="POST" action="/credits/transfer">
        <p>Send credits to: </p><input type="text" name="to"><br />
        <p>Amount: </p><input type="number" name="amount"><br />
        <input type="button" onclick="submitForm()" value="Submit">
    </form>
    <script>
        function submitForm() {
            var form = document.forms["transfer"];
            var xhr = new XMLHttpRequest();
            xhr.withCredentials = true; //send cookies
            xhr.open(form.method, form.action, true);
            xhr.setRequestHeader('X-XSRF-TOKEN', 'nocsrfhere');
            xhr.addEventListener("load", function() {
                alert("Form Submitted!");
            });
            xhr.send(new FormData(form));
        }
    </script>
</body>
```

The custom request header display as following format : X-header-name: value

```
POST /credits/transfer HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0) Gecko/20100101 Firefox/72.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-XSRF-TOKEN: nocsrfhere
Content-Type: multipart/form-data; boundary=---------------------------2995119424827
Content-Length: 242
Origin: http://localhost
Connection: close
Referer: http://localhost/safe_form_custom_header.html
Cookie: SESSIONID=3345758109329575839209178I

---------------------------2995119424827
Content-Disposition: form-data; name="to"

batman
---------------------------2995119424827
Content-Disposition: form-data; name="amount"

1000
---------------------------2995119424827--
```

Server only needs validate if that custom header there. If it is, the request must have come from Javascript and therefor same origin

Reference:

OWASP:https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html

Seclab : https://seclab.stanford.edu/websec/csrf/csrf.pdf

Speakerdeck: https://speakerdeck.com/ropnop/dont-cross-me-same-origin-policy-and-all-the-cross-vulns-xss-csrf-and-cors?slide=43