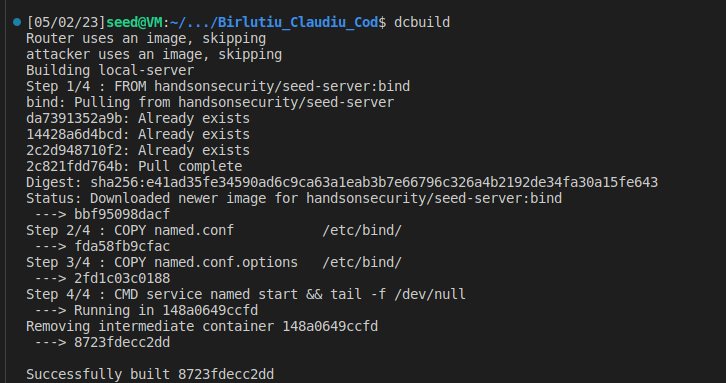
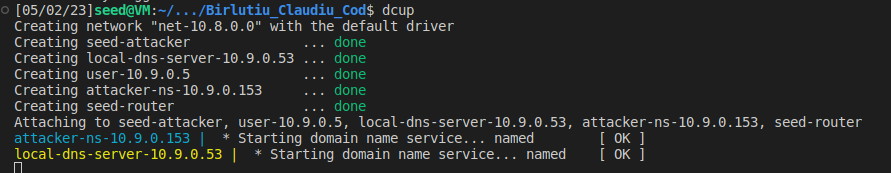
# Raport pentru lucrarea 8: Atacul asupra DNS local

# Autor: Birlutiu Claudiu-Andrei

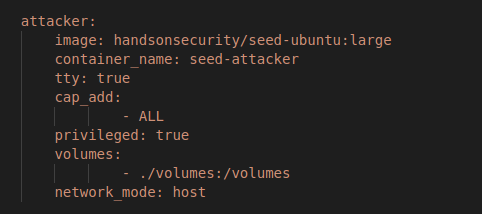
# Setarea mediului

* Pentru a crea mediul local de simulare a atacului DNS a fost nevoie de crearea celor 4 containere docker: unul pentru victima, unul pentru serverul DNS local și 2 pentru atactor
* am rulat comand dcbuild urmat de dcup pentru construirea containerelor necesare și pornirea lor





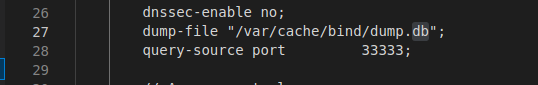
* de remarcat este ca la configurarea container-ului pentru atacator avem setat **network\_mode** la **host** pentru a putea vedea pachetele din alte containere



* în acest caz, containerul attacker este configurat astfel încât vede toate interfetele de retea ale gazdei si chiar areaceleasi adrese IP ca si gazda.

În ceea ce prveste containerul pentru DNS:

* observam ca în fișierul **name.conf.options** avem fixat portul sursa 33333



* de asemenea observam ca a fost dezactivat sistemul de protecția **dnssec**
* am observat de asemenea în fișierul **name.conf** configurarea domeniului attacker32.com, car în momentul în care este accesat se va face o redirectare spre serverul de domeniu din containerul atacatorului

ASPECTE CONTAINER USER

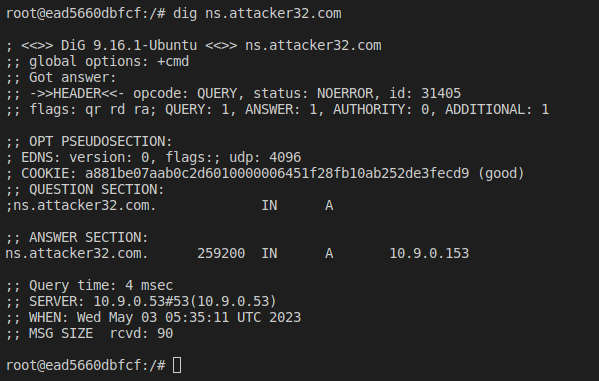
* în fișierul resolv.conf regasim faptul ca serverul 10.9.0.53 este adaugat primul ca server de nume și va juca rolul de dns -server

ASPECTE CONTAINER SERVER NUME ATACATOR

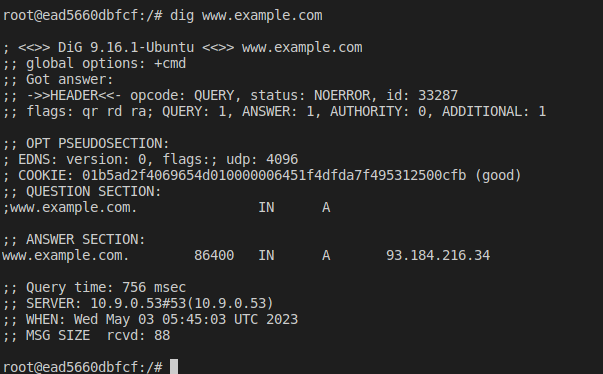
* în acesta în fișierul named.conf sunt declarate 2 zone:
  + - zona legitima a atacatorului **attacker32.com**
    - zona falsa **example.com**

TESTARE CONFIGURARI

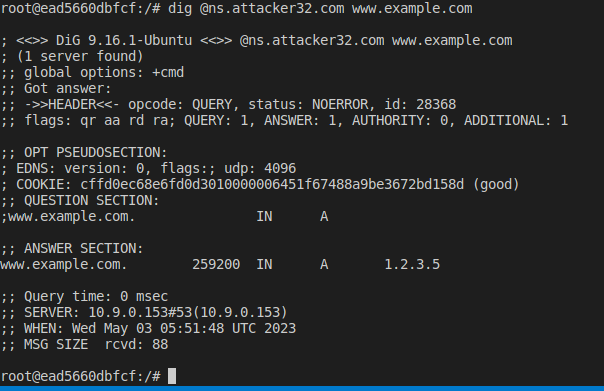
* m-am conectat la containerul user; am rulat comanda *dig ns.attacker32.com* și am obitnut rezultatele din
* comanda dig se folosește pentru a interoga serverele DNS și pentru a obține informații despre adresele IP, recordurile DNS și alte informații de rețea.
* Deoarece în serverul dns e configurat faptul să se facă o redirectionare spre dns server al atacatorului -> obținem informații din fiserul attacker32.com.zone



* în continuare am rulat comand *dig* [*www.example.com*](http://www.example.com/)și am observat faptul ca obținem un ip public, deci cererea a fost trimisa către serverul oficial de nume al domain name-ului example.com



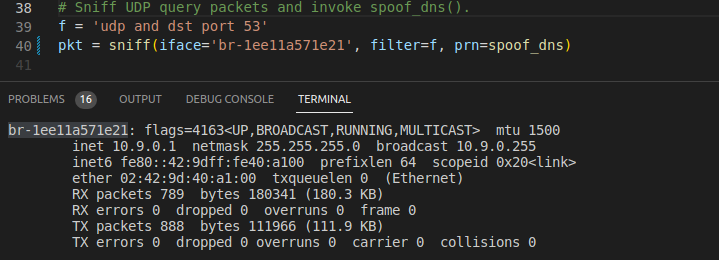
* dacă punem *dig @ns.attacker32.com* [*www.example.com*](http://www.example.com/)
  + interoghează serverul DNS specificat prin "ns.attacker32.com" pentru a obține informații despre adresa IP asociată cu domeniul "[www.example.com](http://www.example.com/)"
  + obținem ip-ul din zone\_example.com din serverul de nume al atacatorului



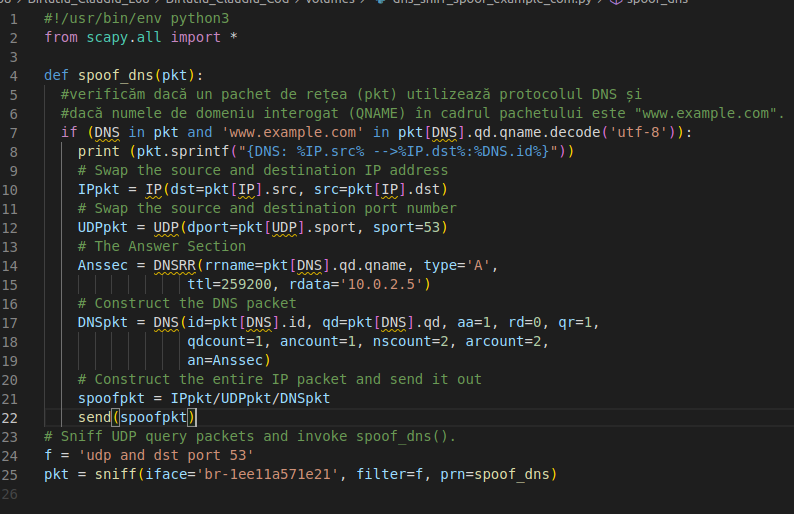
# Sarcini de atac

## Sarcina 1: Falsificarea directă a răspunsului dat utilizatorului

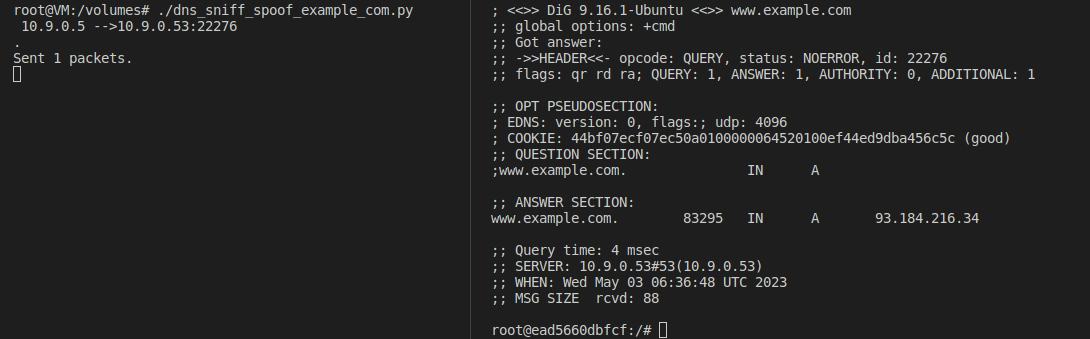
* In prima faza m-am conectat pe containerul atacatorului si am rulat ifconfig pentru a vedea interfata pentru 10.9.0.0 pentru a modifica fisierul dns\_sniff\_spoof.py



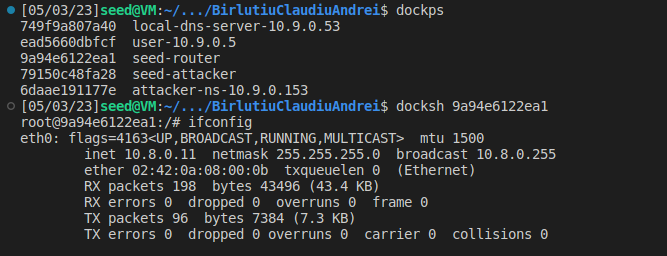
Fisierul de atac arata in felul urmator; am preluat informatia din fisierul dns\_sniff\_spoof.py pus la dispozitie in laborator si am facut pentru example.com



* rulam programul dns\_sniff\_spoof\_example\_com.py creat (cel de sus) din containerul atacatorului
* in prima faza observam ca atacatprul a interceptat cererea DNS si chiar daca a trimis un pachet ca raspuns, userul a primit raspuns mai rapid de la serverul de nume real al domeniul [www.example.com](http://www.example.com/)



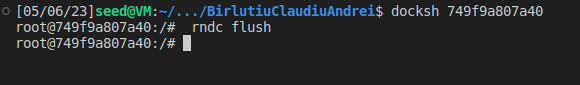
* pentru a remedia problema, am aplicat sugestiile din laborator prin care incetinesc traficul de internet
  + - gasim in prima faza interfata pentru 10.8.0.0 si aceasta e eth0

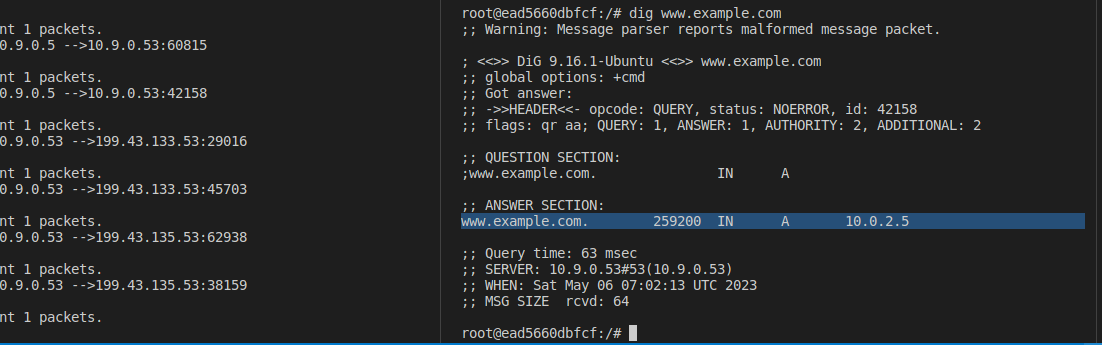


* + - al doilea pas este executia comenzii tc pe eth0: *tc qdisc add dev eth0 root netem delay 100ms*



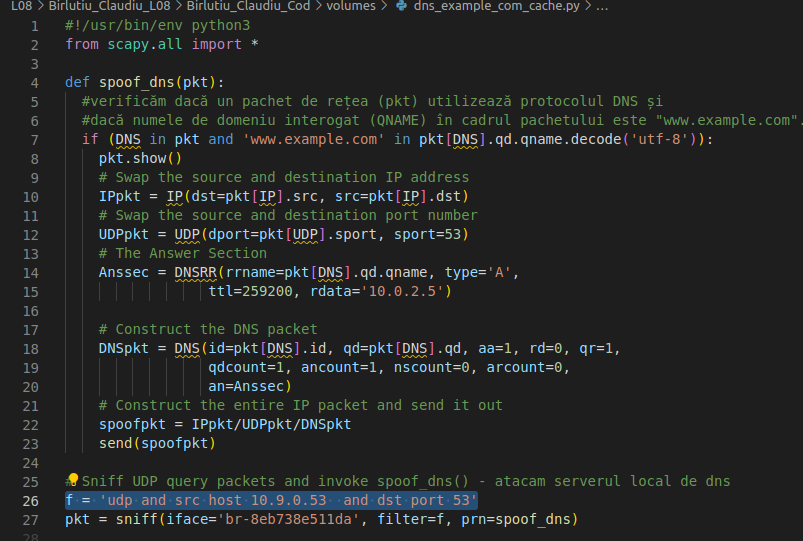
* + - facem clear la dns-ul local



* + - executam comanda dq si observam ca nu se va mai ajunge la serverul de nume real al domeniului example.com ci la atacator

## Sarcina 2: Atacul cu otrăvirea cache DNS – falsificarea răspunsurilor

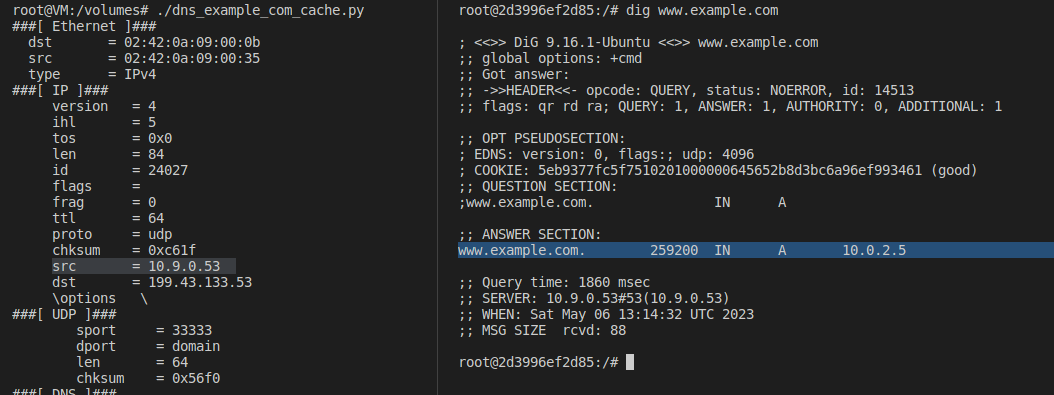
* Dezavantajul solutiei anterioare este faptul ca de fiecare data ca nd masina utilizatorului trimite o interogare DNS pentru domeniul [www.example.com](http://www.example.com/), masina atacatorului trebuie sa trimita DNS-ul falsificat – metoda ineficienta
* o metoda mult mai buna este de a ca tinit serverul DNS, in locul containerului utilizator; severul DNS cauta prima data in cache, iar apoi va incerca sa gaseasca raspunsuri de la alte servere DNS → atacatorul poate falsifica raspunsul de la alte servere DNS, iar serverul local va pastra in cache acest raspuns
  + - atacatorul va putea sa falsifice doar o data, iar serverul local va pastra in cache informatia pana cand este setata sa expire → atacul se numeste otravirea cache DNS
* am modificat filtrul pentru capturarea cererilor DNS din partea server-ului de nume local DNS care ruleaza pe **10.9.0.53** se poate observa in fisierul **dns\_example\_com\_cache.py**



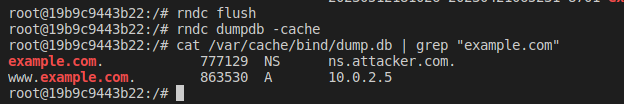
* am curatat cache-ul din server-ul local de DNS



* am rulat scriptul **dns\_example\_com\_cache.py** pe containerul atacatorului si am rulat comanda dig [www.example.com](http://www.example.com/) pe containerul user pentru a vedea daca a fost pacalit si observam ca ip—ul returnat e cel din query

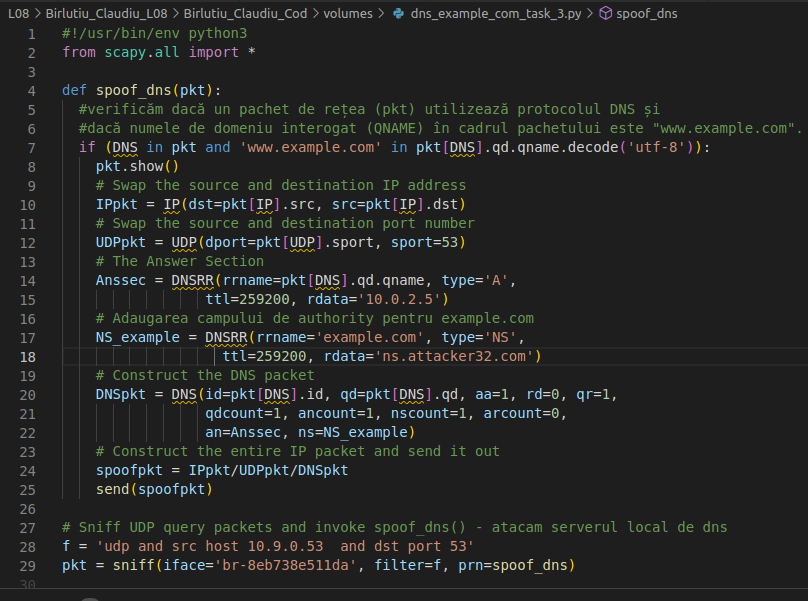


* am verificat cache-ul de pe serverul local de dns si observam ca a fost adaugata intrarea in cache. Astfel, la rularile urmataoare a comenzii dig am observat ca nu se va mai ajunge sa se ajunga sa faca interogare serverul de nume local in exterior, ci se va uita in cache

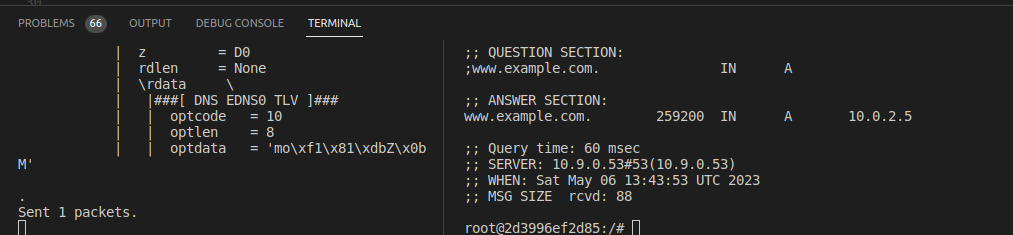


## Sarcina 3: Falsificarea înregistrărilor NS

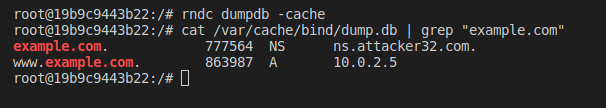
* Atacul anterior afecta doar un un nume gazda, ([www.example.com](http://www.example.com/)), pentru a lansa un atac care poate afecta intregul domeniu example.com vom adauga un nou header in raspunssul atacatorului care include si authority section
* lansarea unui astfel de atac consta in adaugarea **ns.attacker32.com** ca server de nume care va fi folosit pentru interogari viitoare ale oricarui nume de gazda din domeniul **example.com**
* am creat fisierul *dns\_example\_com\_task\_3.py* unde am adaugat si authority section



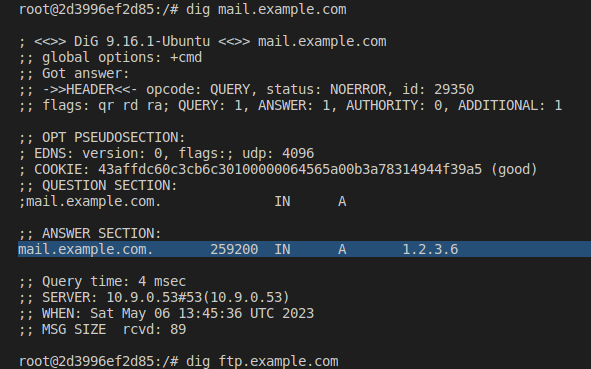
* am facut clean la memoria cache a serverului local de DNS si am exectuat codul de atac pe containerul ataactorului si am rulat comanda dig pe containerul user



* si observmam ca in cache avem adaugat domeniul example.com iar serverul de nume atasat este ns.attacker32.com

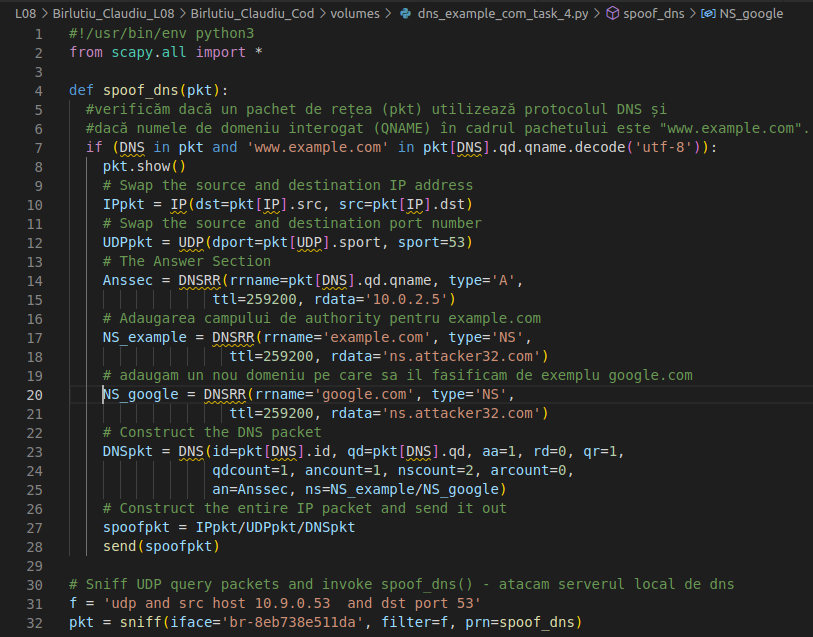


* am lansat in executie si comanda mail.example.com si a observat ca s-a interogat serverul de nume al atacatorului (ns.attacker32.com)



## Sarcina 4: Falsificarea înregistrărilor NS pentru alt domeniu

* Am reusit inainte sa setez ns.attacker32.com ca server de nume pentru domeniul **example.com**
* in *dns\_example\_com\_task\_4.py* am facut modificarile corespunzatoare pentru un atac prin care falsificam si serverul de nume pentru al domeniu cum ar fi cel pentru google.com



* stergem cache-ul din serverul de nume local, lansam in executie scriptul *dns\_example\_com\_task\_4.py* si exceutam interogari de tipul dig pe containerul user



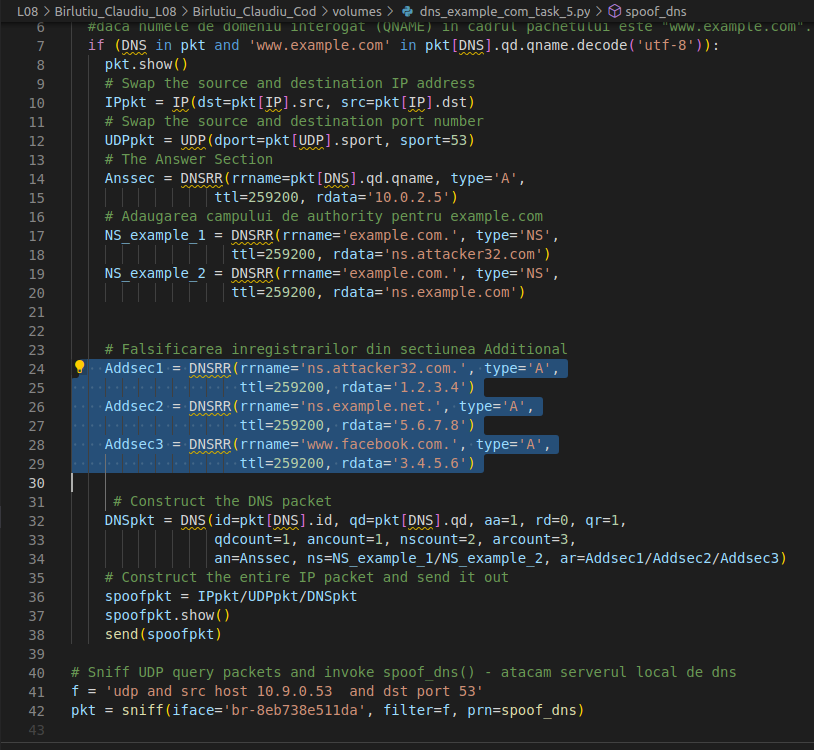
* observam ca s-a pus si domeniu; google.com la serverul de nume ns.attacker32.com



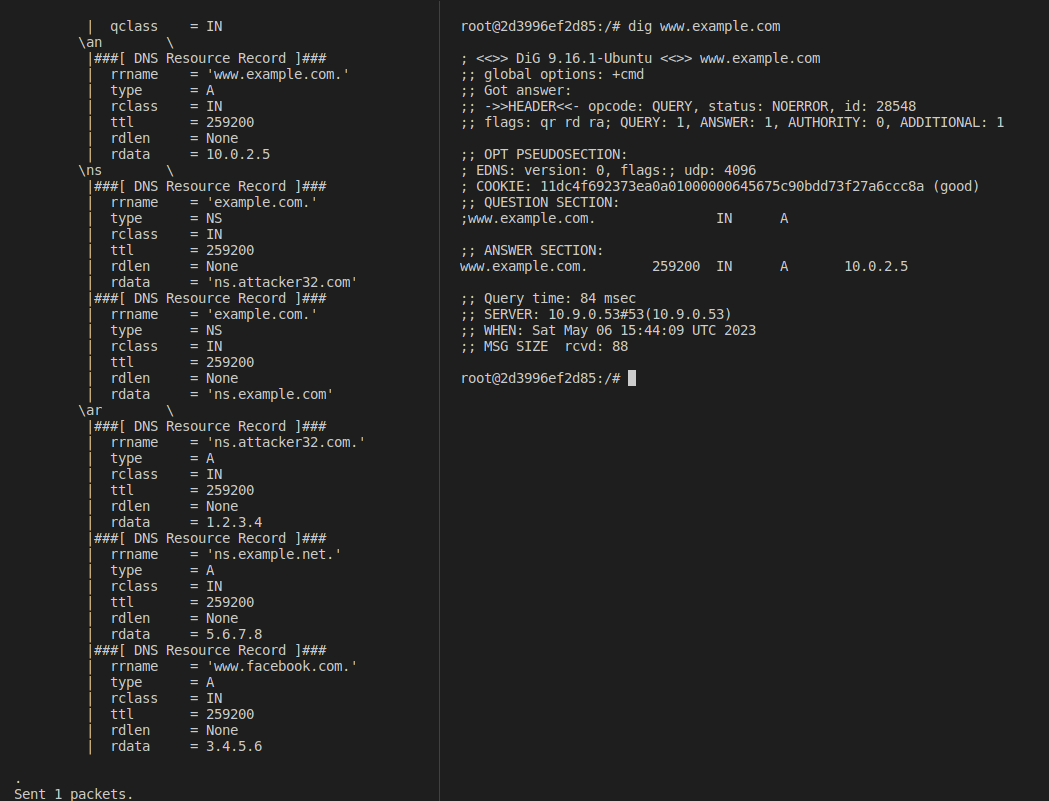
* NU A MERS pur si simplu, doar prin inversare

## Sarcina 5: Falsificarea înregistrărilor din secțiunea Additional

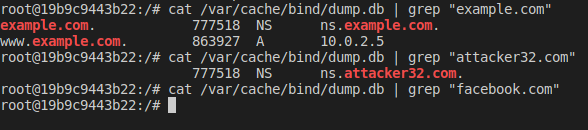
* In cadrul acestei sarcini incercam sa adaugam un additional section cu niste valori pe care dorim sa le punem in DNS cache ( de exemplu sa falsificam serverul de nume pentru facebook.com ) - o sa vedem ca nu se poate
* am creat urmatorul scritp python cu aditional section-ul mentionat



* am sters cache-ul din serverul local de DNS si am executat scriptul de mai sus in containerul atacatorului; aceeasi interogare pusa in containerul user (dig [www.example.com](http://www.example.com/))



* cautam in cache daca s-au adaugat valorile dorite



* ceea ce s-a adaugat este la domeniul **example.com** cele 2 servere de nume ce pot fi interogate pentru a accesa ip-ul domeniului ( ns.example.com su ns.attacker.com) deoarece acestea au fost adaugate in sectiunea de authority
* in schimb, toate cele 3 intrari pe care le-am dorit sa le adaugam in cache (attacker32.com., ns.example.net. Si [www.facebook.com](http://www.facebook.com/)) nu s-au salvat in cache, ci au fost ignorate, doarece nu faceau referire la domeniul example.com