CNS LAB

LAB - 5

Remote DNS cache Poisoning Attack Lab

NAME : SIRI S

SEMESTER : 5

SECTION :H

SRN : PES1UG19CS485

Lab Setup:

ATTACKER: 10. 0. 2. 4

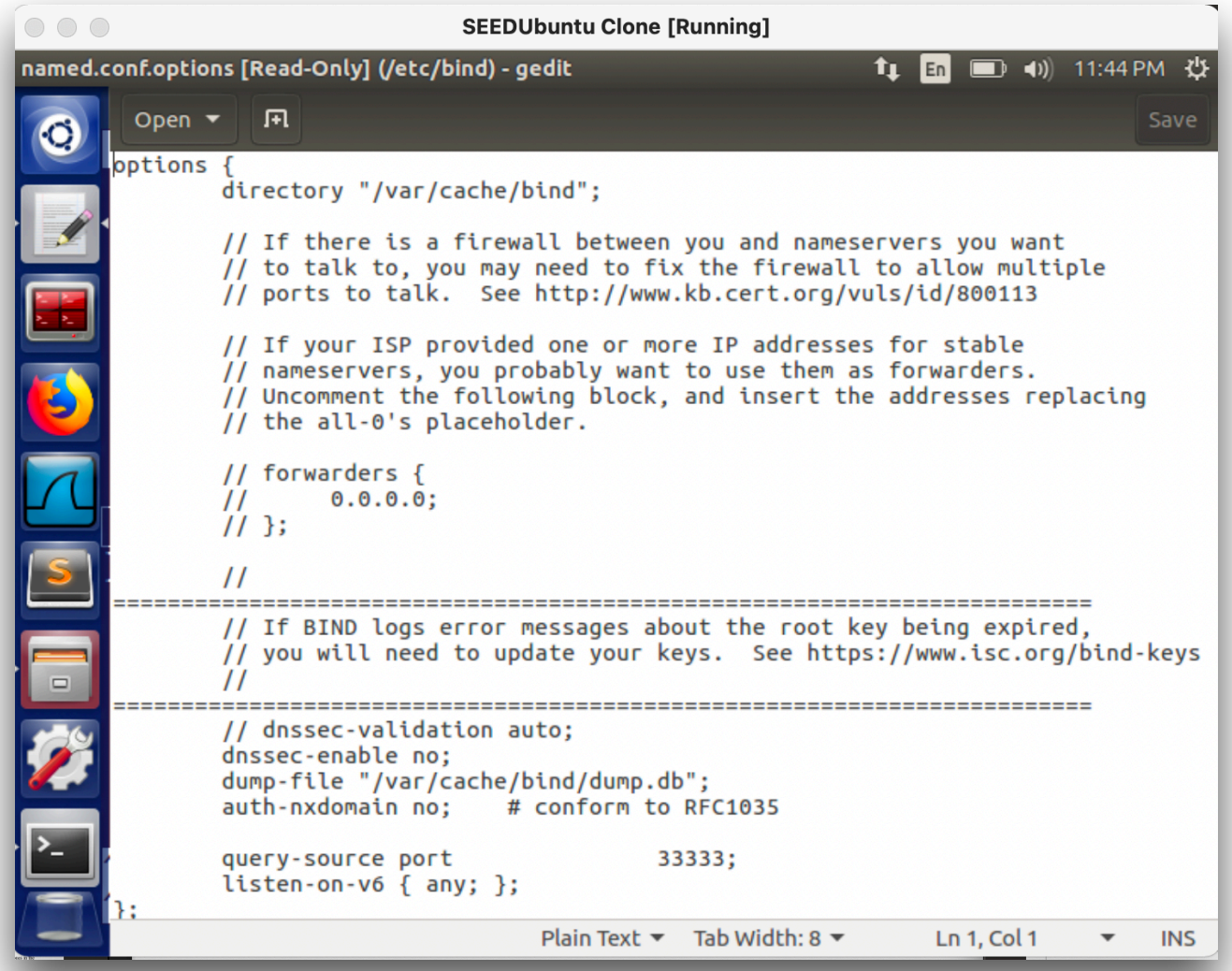
DNS SERVER: 10. 0. 2. 5

USER/VICTIM: 10. 0. 2. 6

Task 1: Configure the Local DNS Server

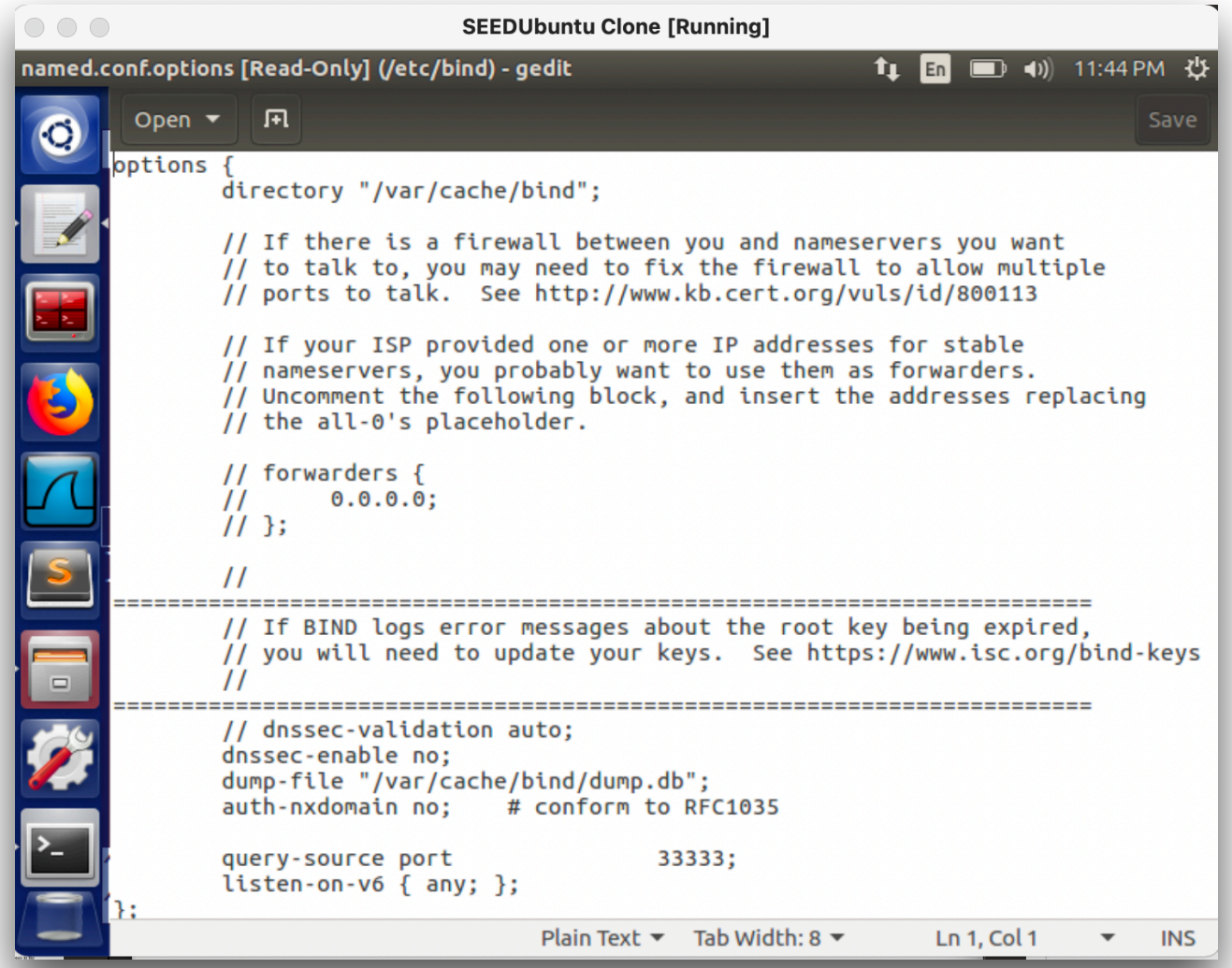
Step 1: Configure the BIND9 Server.

Configure the BIND9 server: This is done by adding the option – "dump-file "/var/cache/bind/dump.db". This basically tells us where the dump file should be stored.



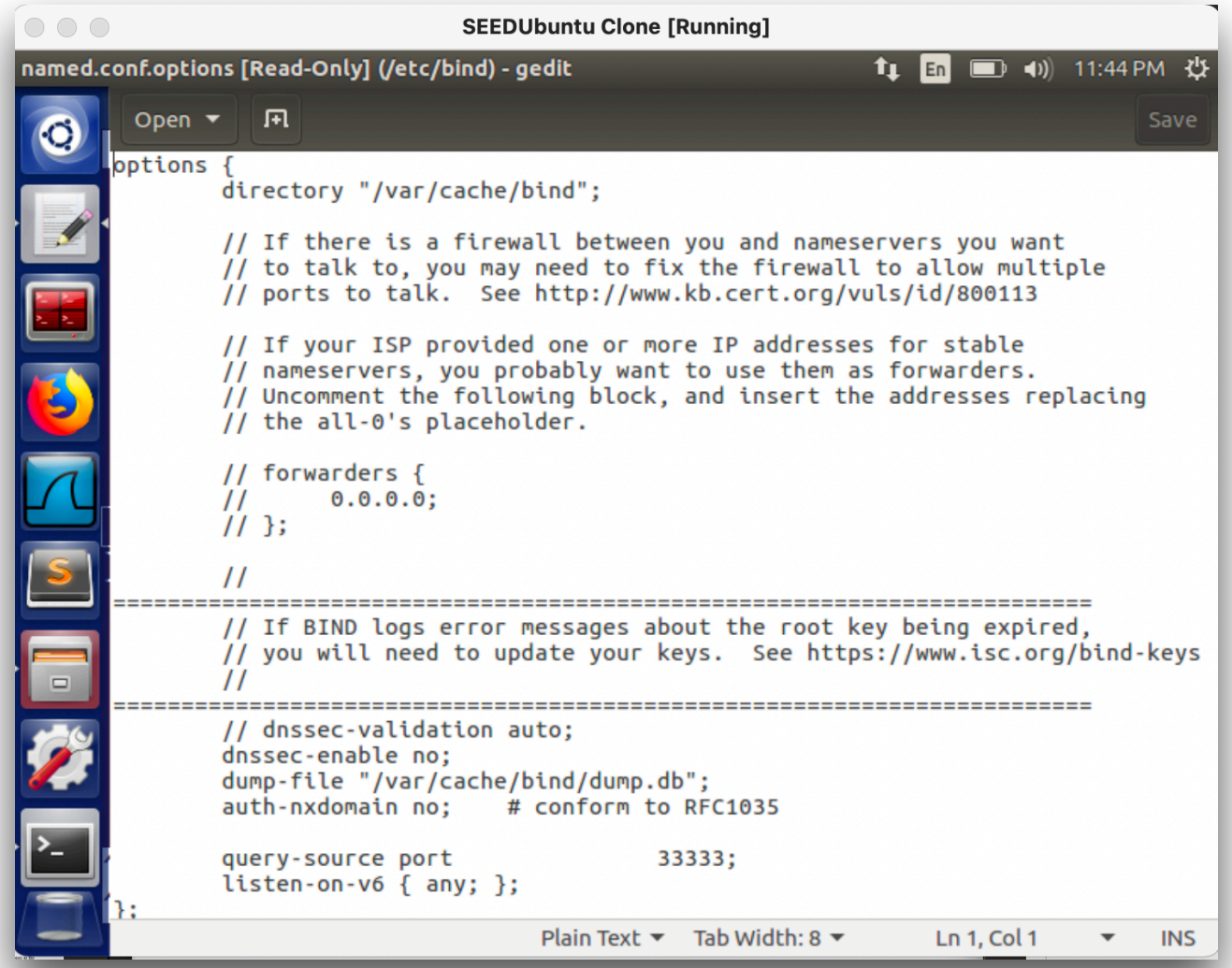
<u>Step 2:</u> Turn off DNSSEC

We set *dnssec enable* to no This switches off the protection against DNS spoofing:



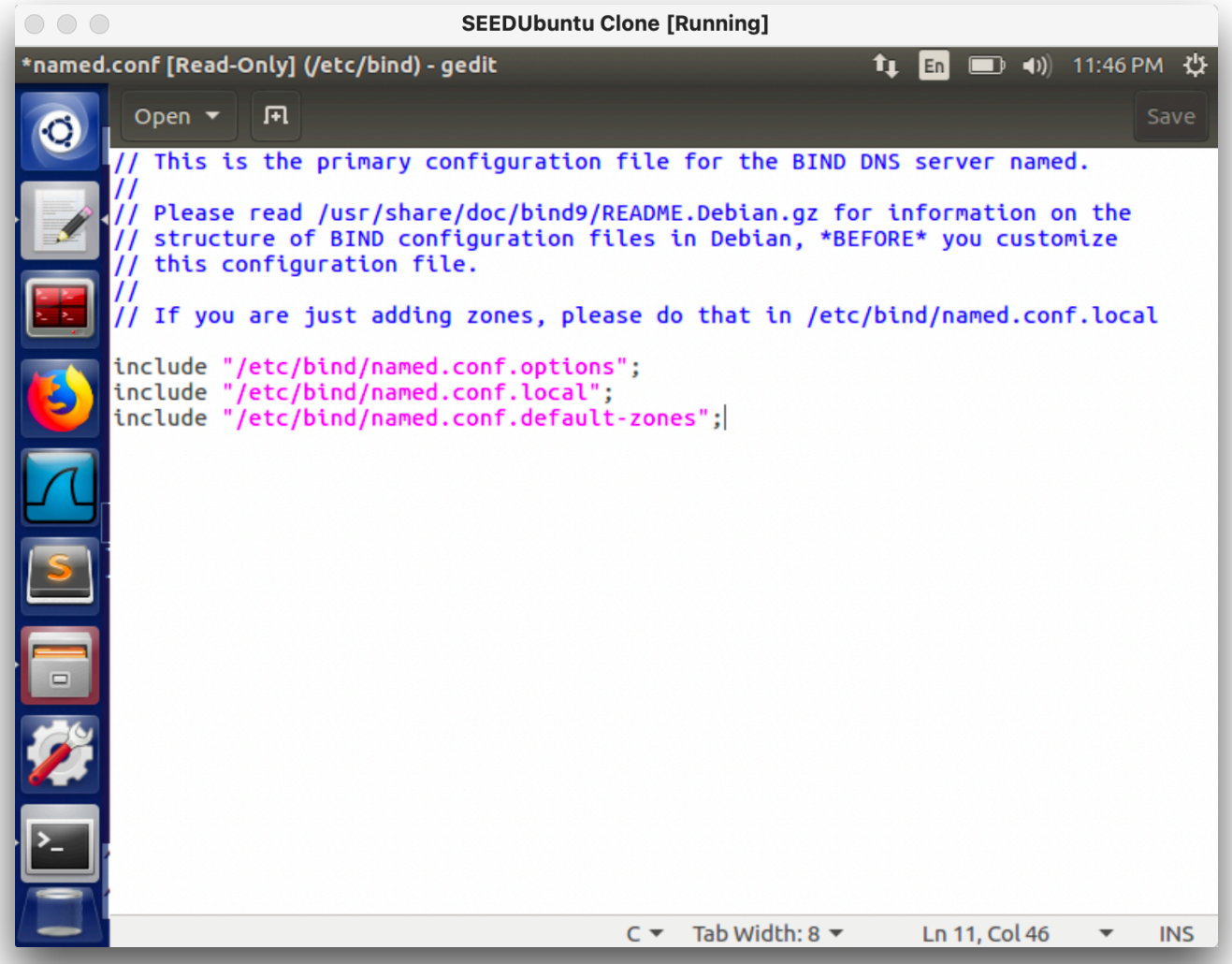
Step 3: Fix the Source Ports

This is done by setting the source port of all DNS queries to 33333.



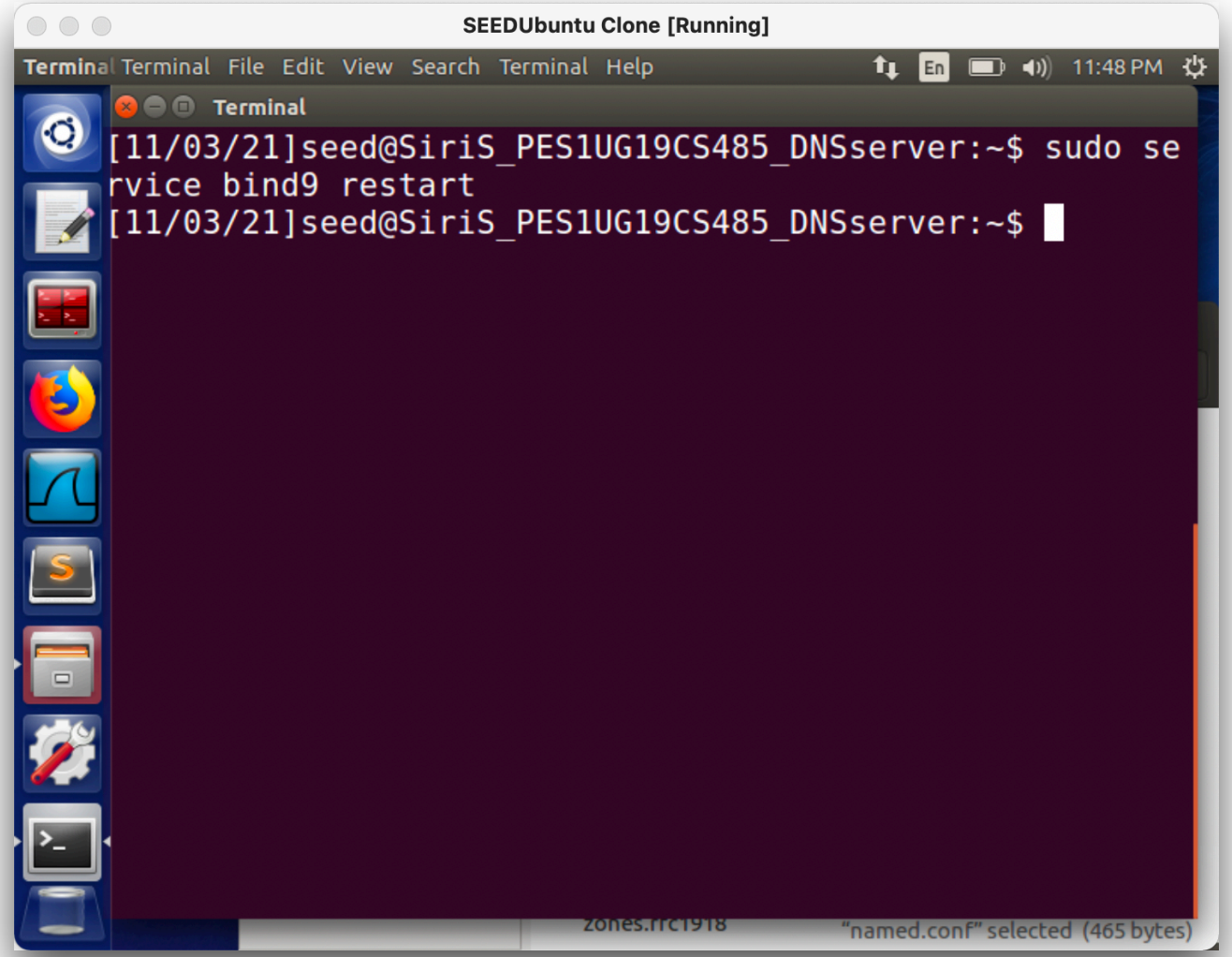
<u>Step 4:</u> Remove the example.com zone

In the previous lab, we configured the local DNS server Apollo to host the example.com domain.In this lab, this DNS server will not host that domain, so we remove it



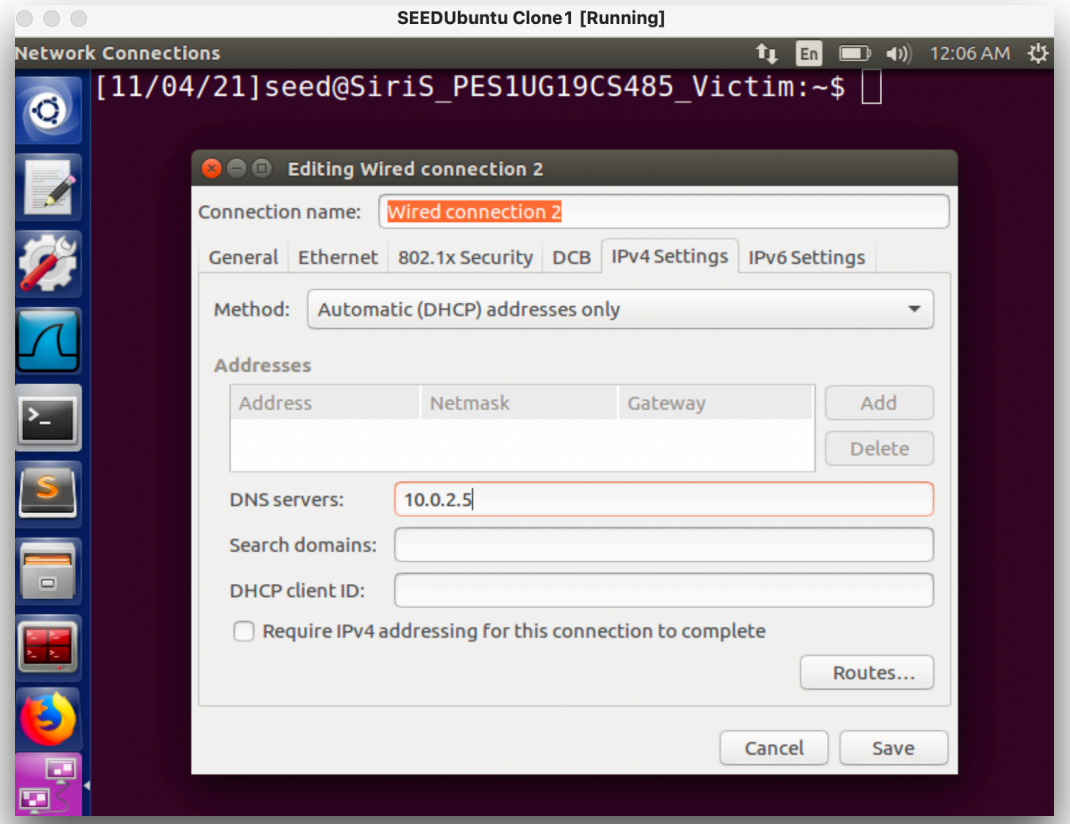
<u>Step 5:</u> Start DNS server

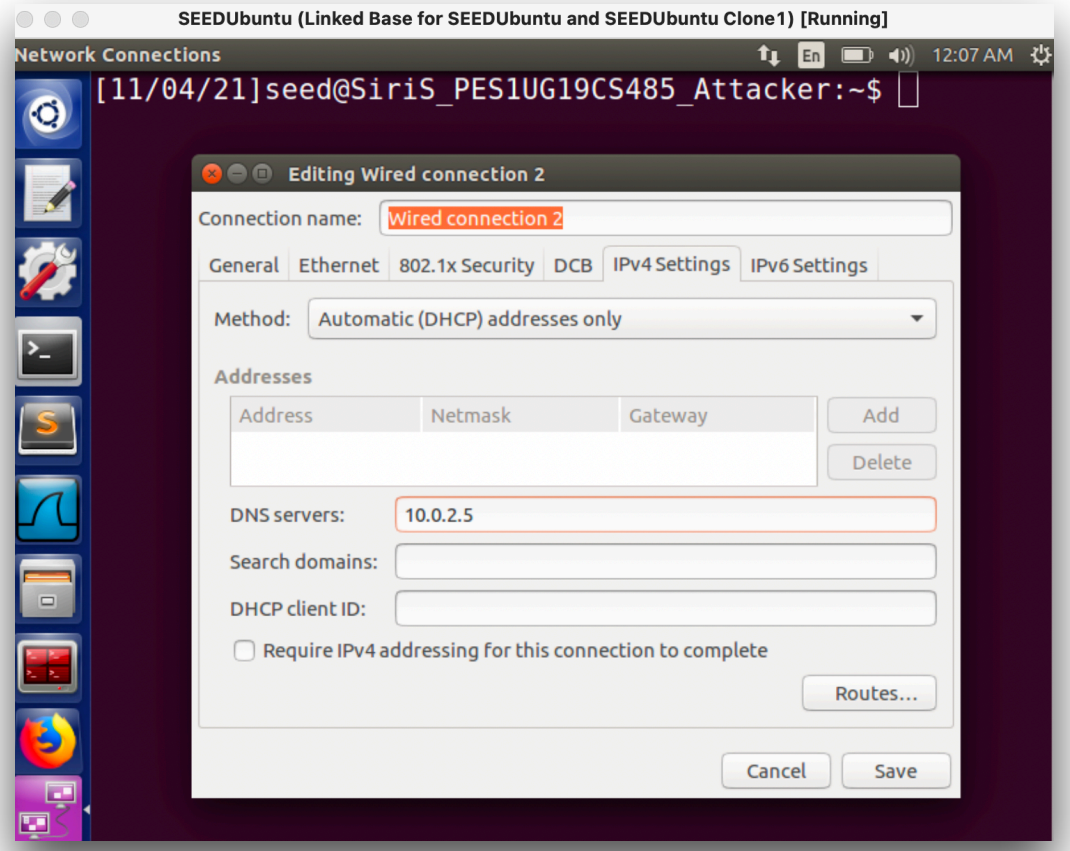
Every time we make a change in the DNS Sever, we need to restart the server in order for the changes to make a difference:



Task 2: Configure the Victim and Attacker Machine

First we open Edit connections and then enter the IP address of our DNS server in the DNS server field of IPv4 settings of both Victim and attacker machines:





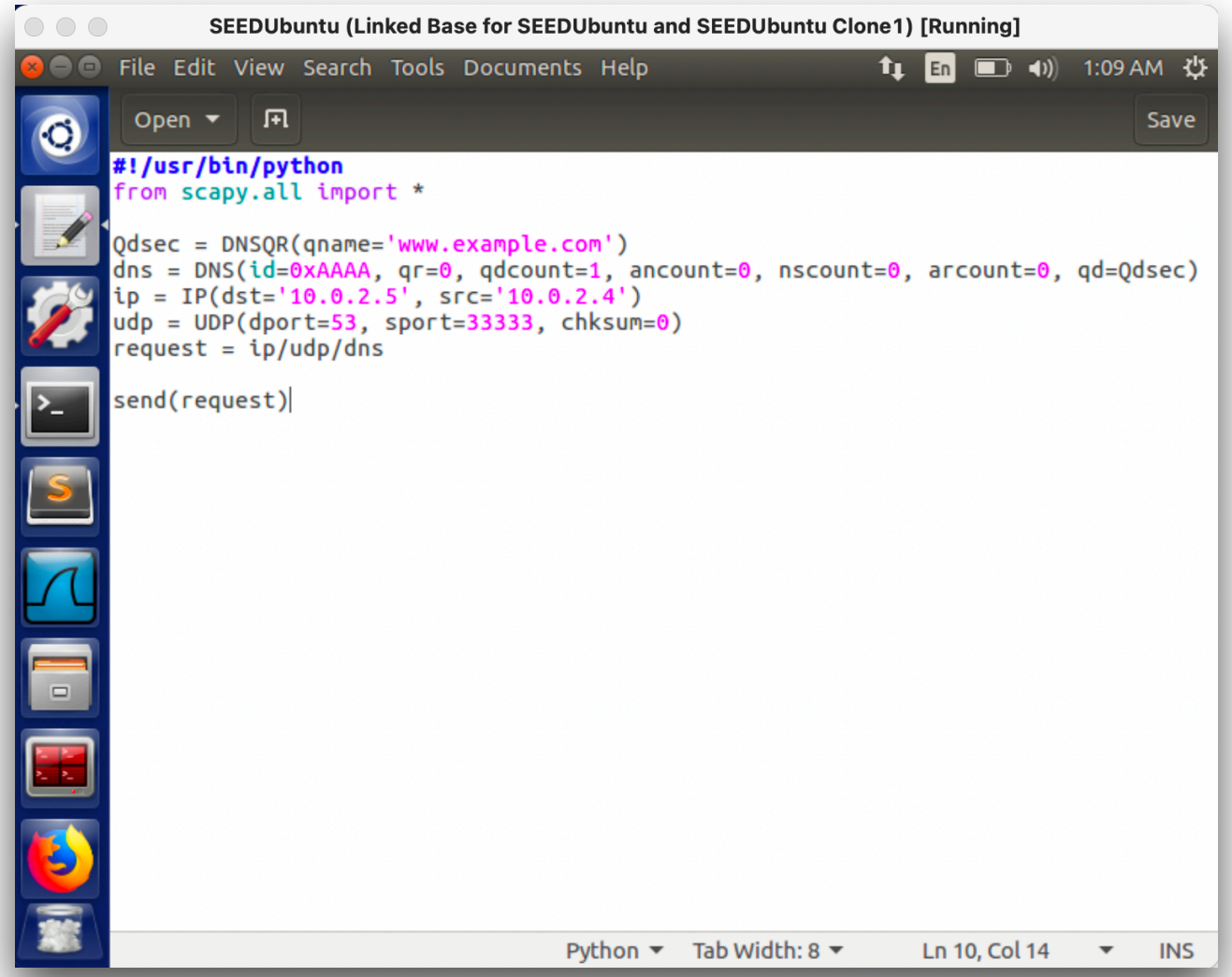
Tasks 3.1: The Kaminsky attack

We configure the attacker machine, so it uses the targeted DNS server as its default DNS Server as its default DNS server. The attacker machine is on the same NAT network. This attack is performed by spoofing DNS Request followed by DNS Replies.

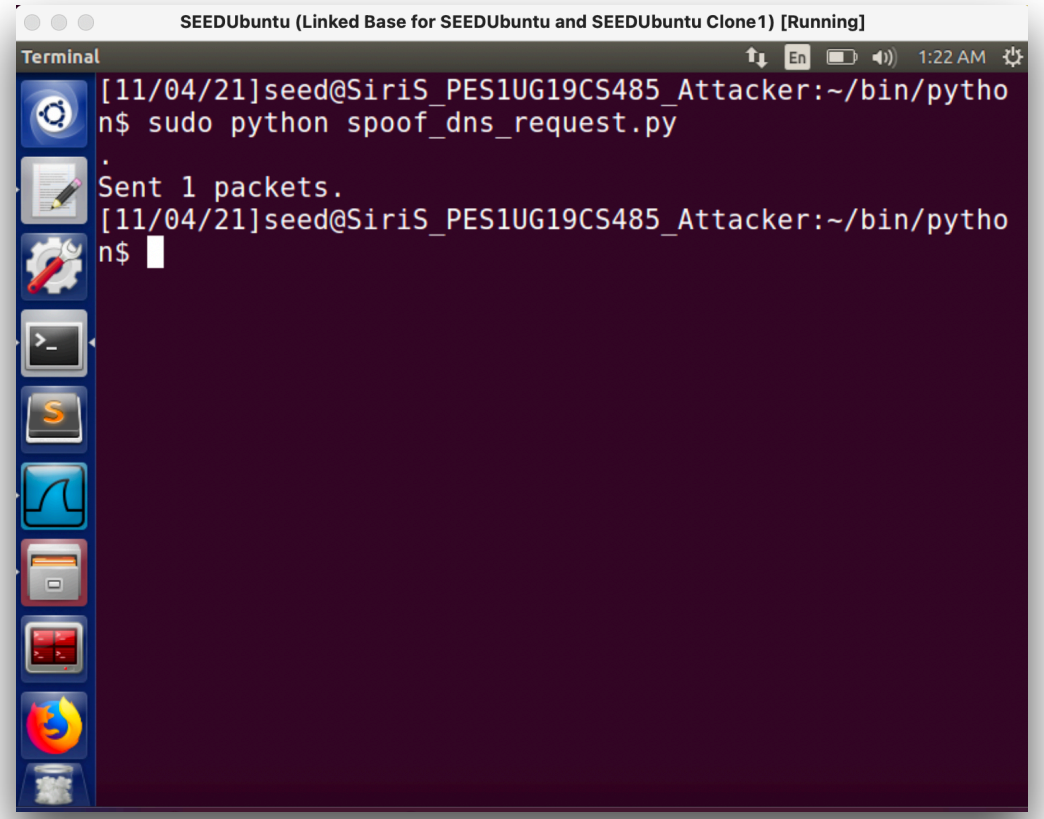
Task 3.1.1: Spoofing DNS Request

We will spoof DNS Requests that trigger the target DNS server to send out DNS queries, so we can spoof DNS replies.

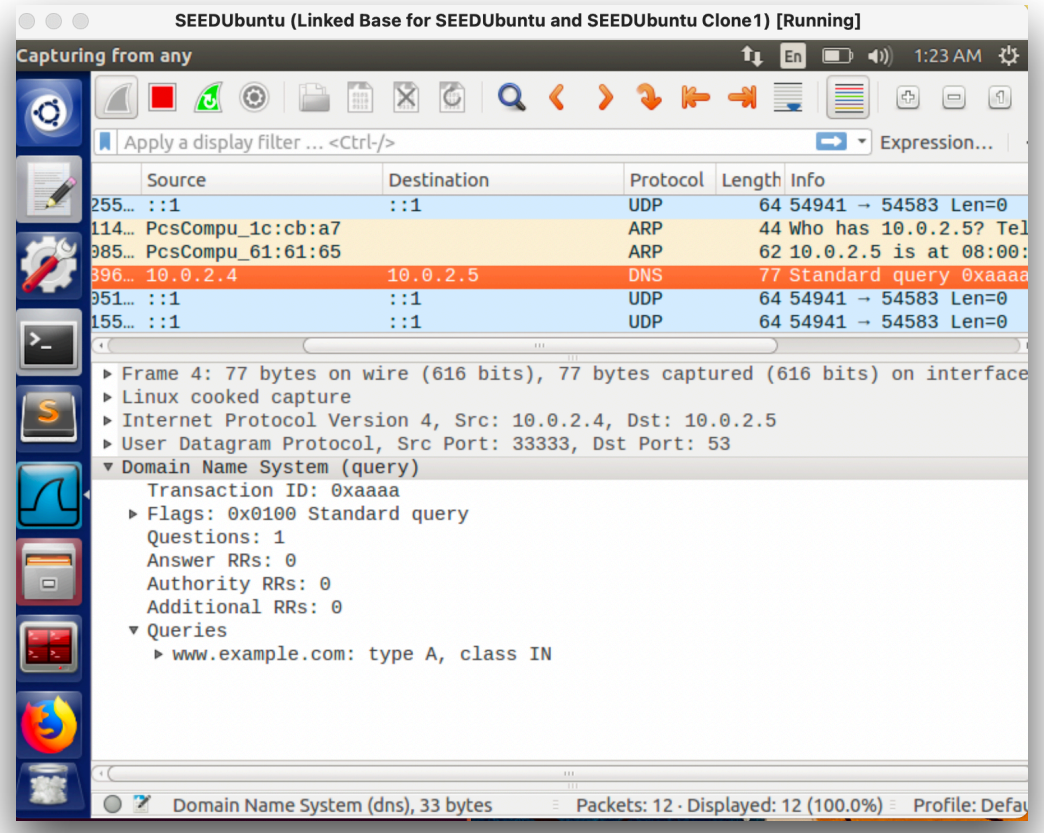
This code constructs a spoofed dns query and sends it:



Upon running the above code:



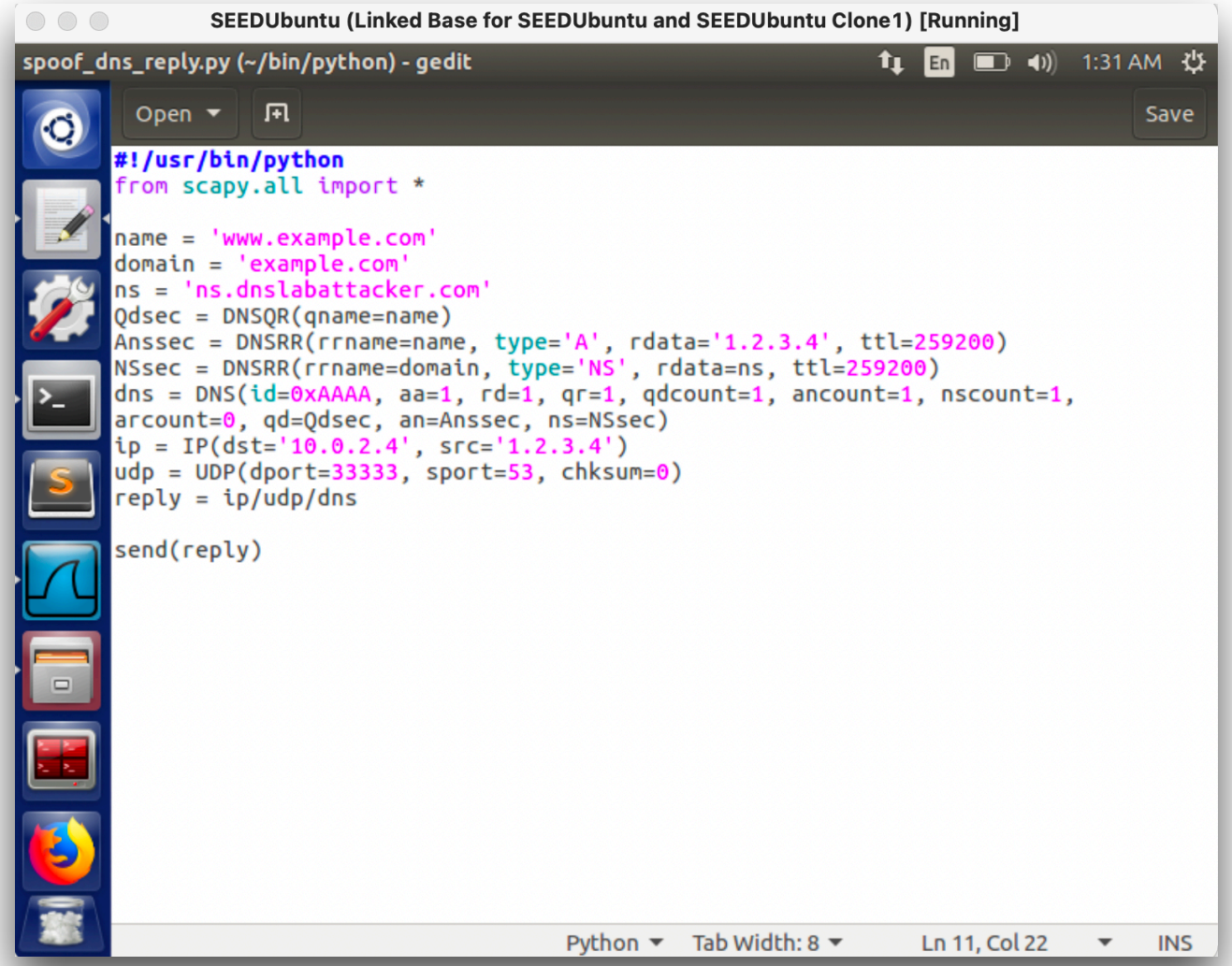
The Wireshark screenshot to show that a response to the dns query was received back from the DNS server:



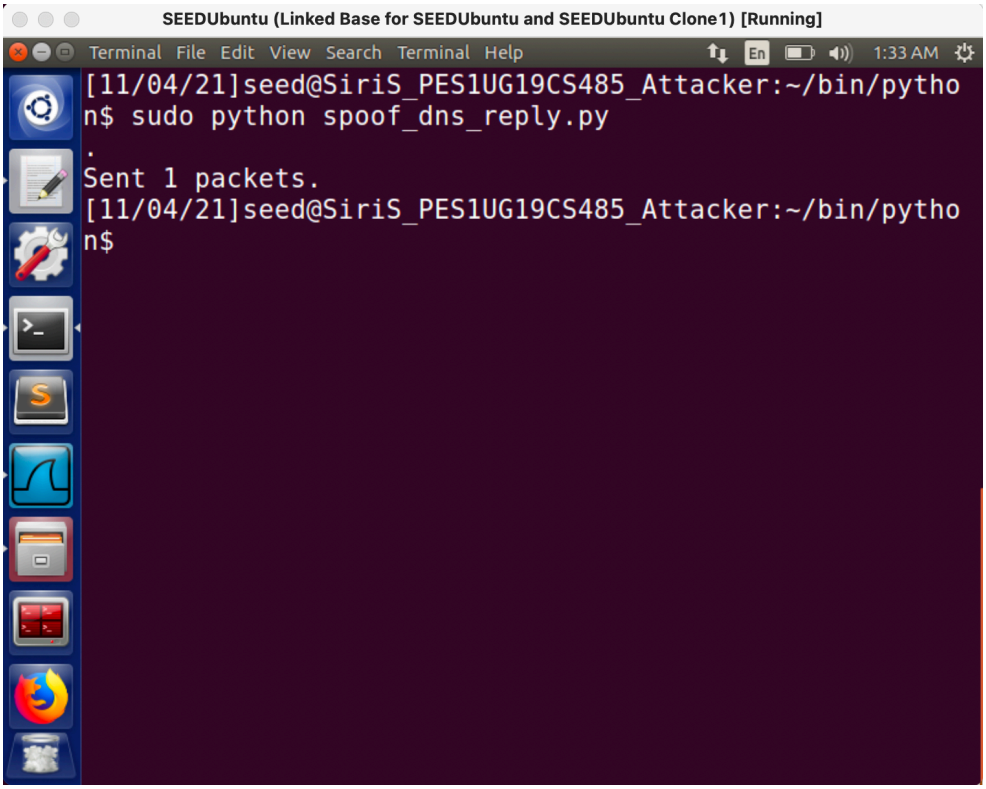
Task 3.1.2: Spoofing DNS Replies

We will spoof DNS Responses to the local DNS Server for each query. We will create a DNS Header with DNS Payload with the Answer, Authority and Additional section.

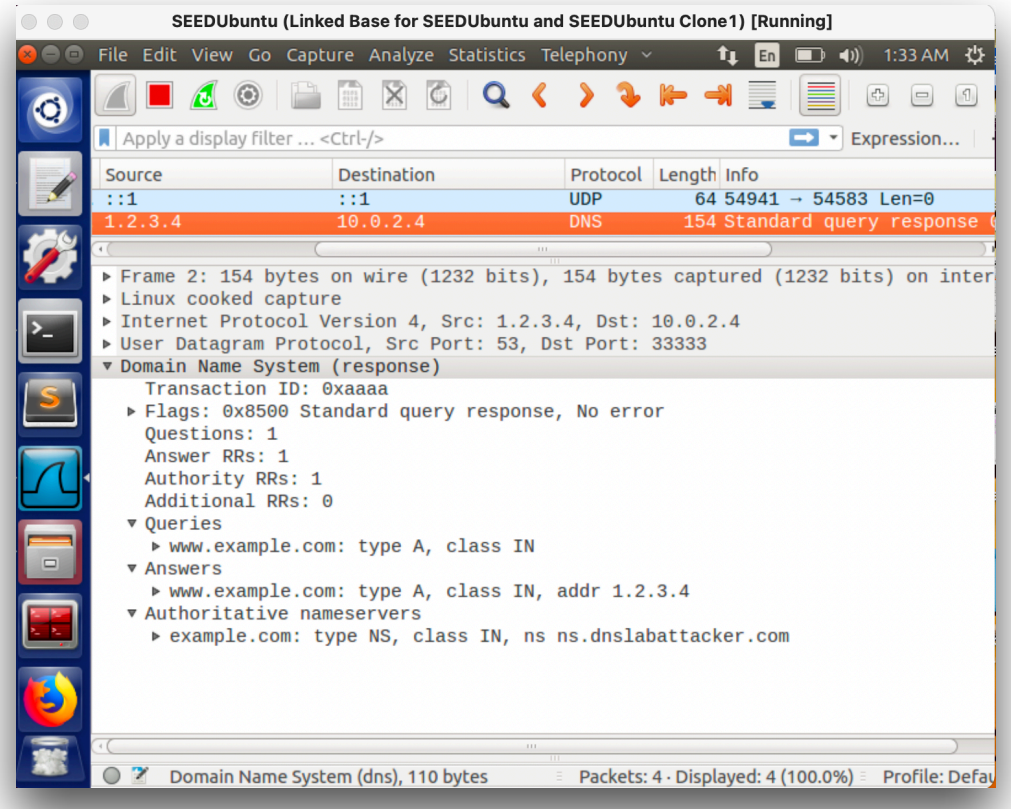
This code constructs a spoofed dns reply and sends it:



Upon running the above code:



The Wireshark screenshot to show that the attacker has received the dns message which was forged



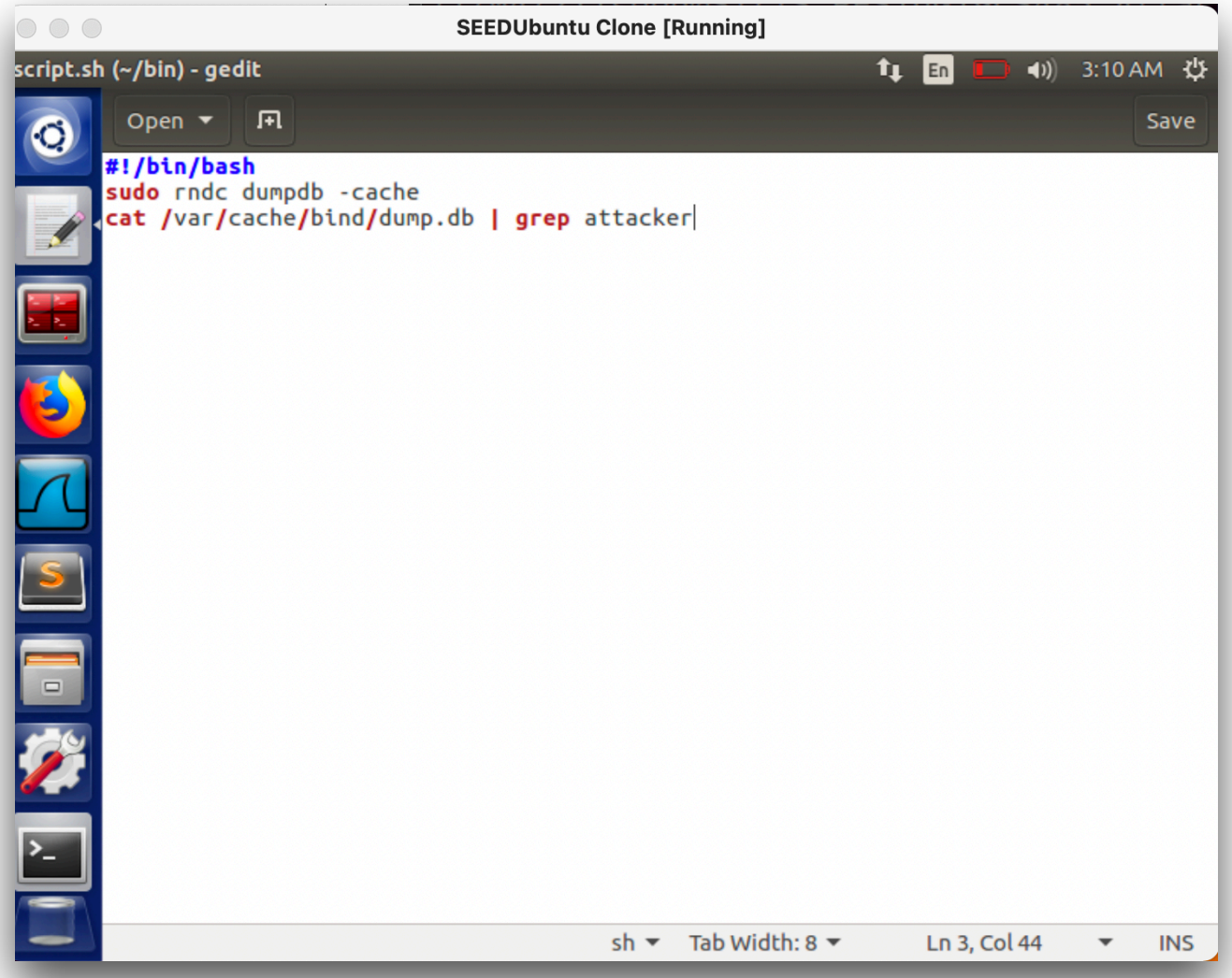
Hence, the spoofing was successful. Now we put everything together to conduct the Kaminsky attack where we send many spoofed DNS replies hoping that one of them hits the correct transaction number and arrives sooner than the actual legitimate replies. The success rate is too low if we stick to Scapy, thus we will use C code for this attack.

Task 3.2: The Kaminsky Attack

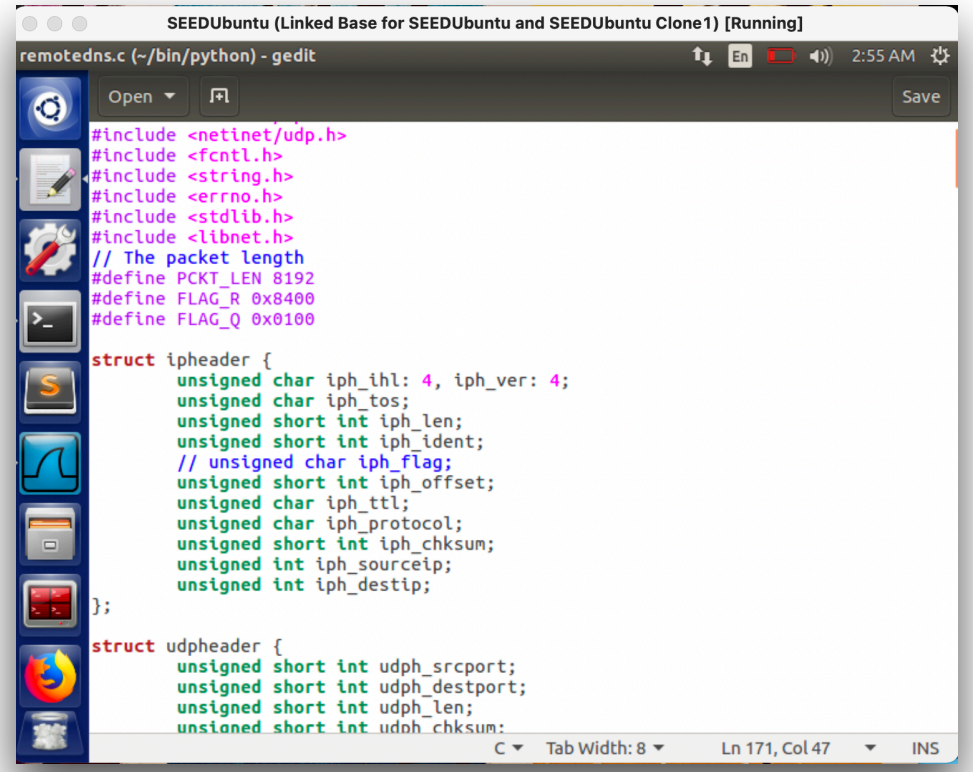
(C code used has been taken from the github repo:

https://github.com/piergiorgioladisa/KaminskyAttack)

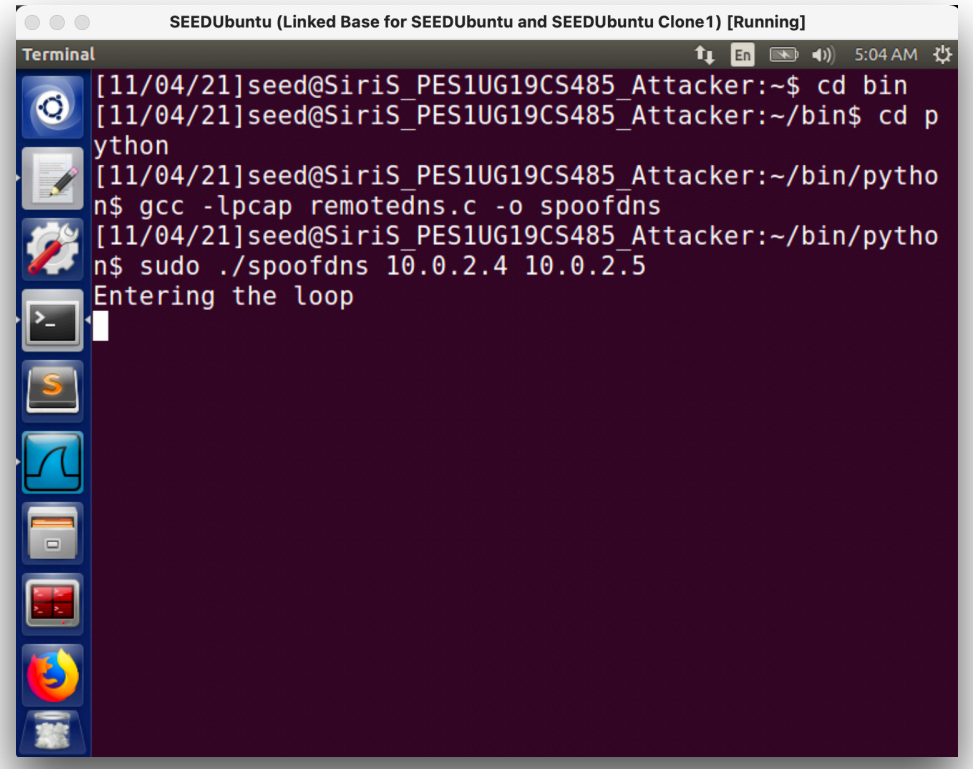
Shell script used on DNS server to check if the cache has been poisoned or not:



The code used:

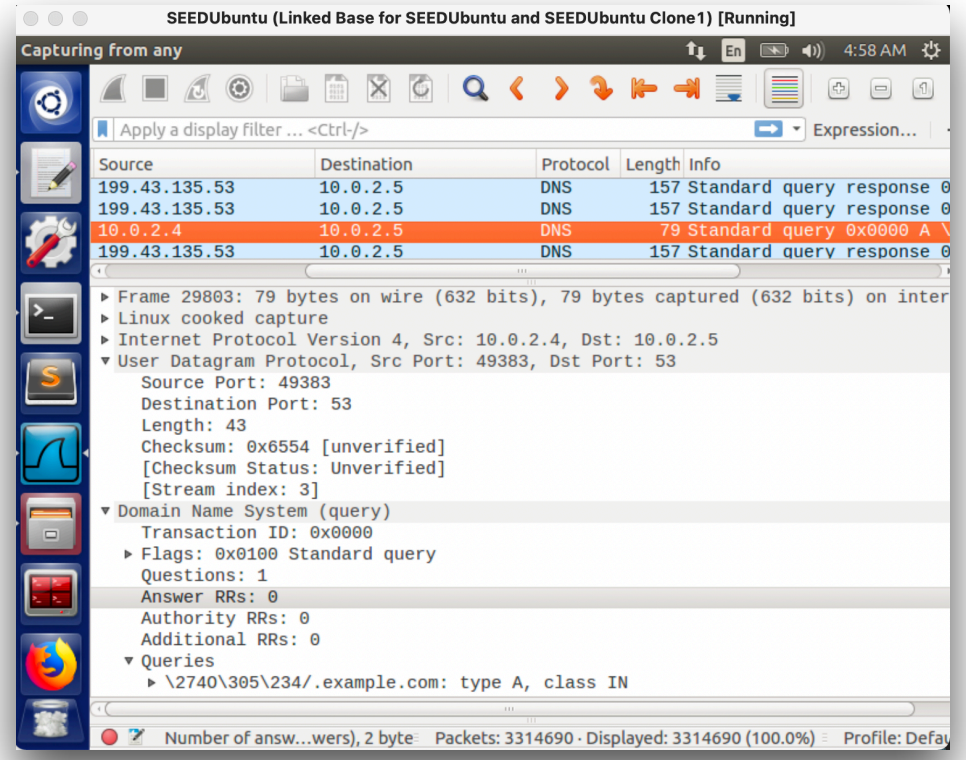


Upon running the code:

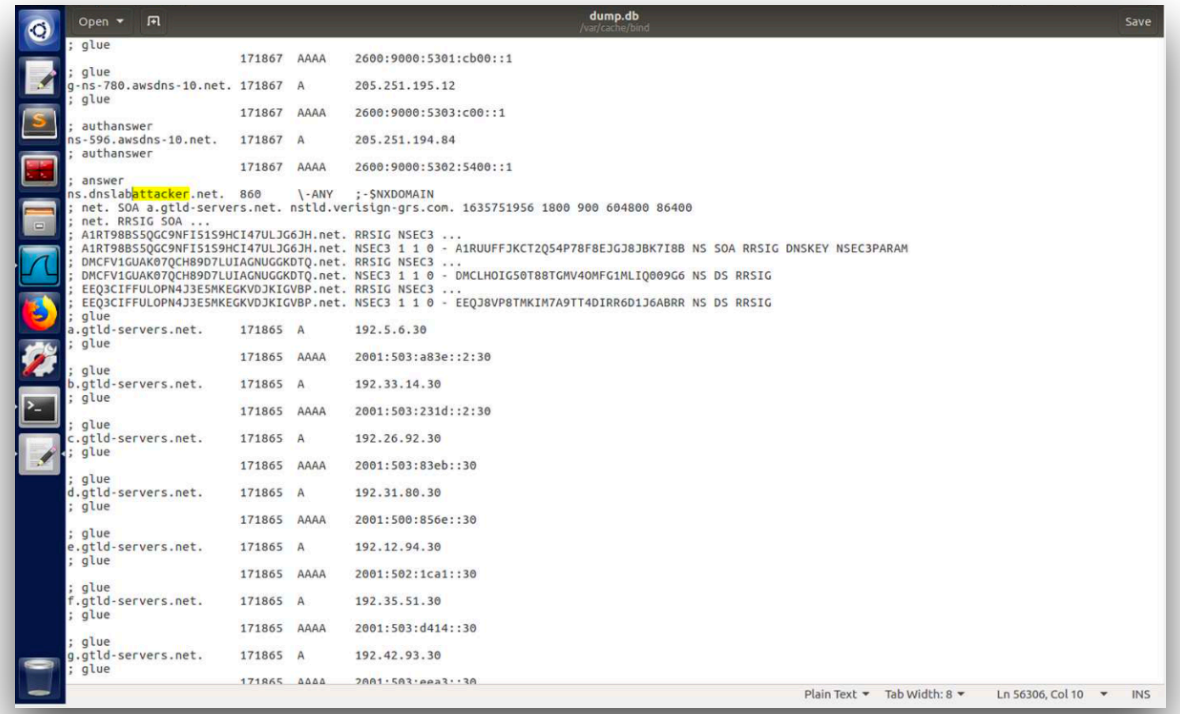


We run the script file multiple times to check if there is a change in the cache:

The Wireshark screenshot:



The dump.db file which shows that the attack was successful, the nameserver for example.com is now set to ns.dnslabattacker.net :



Task 3.3: Result Verification

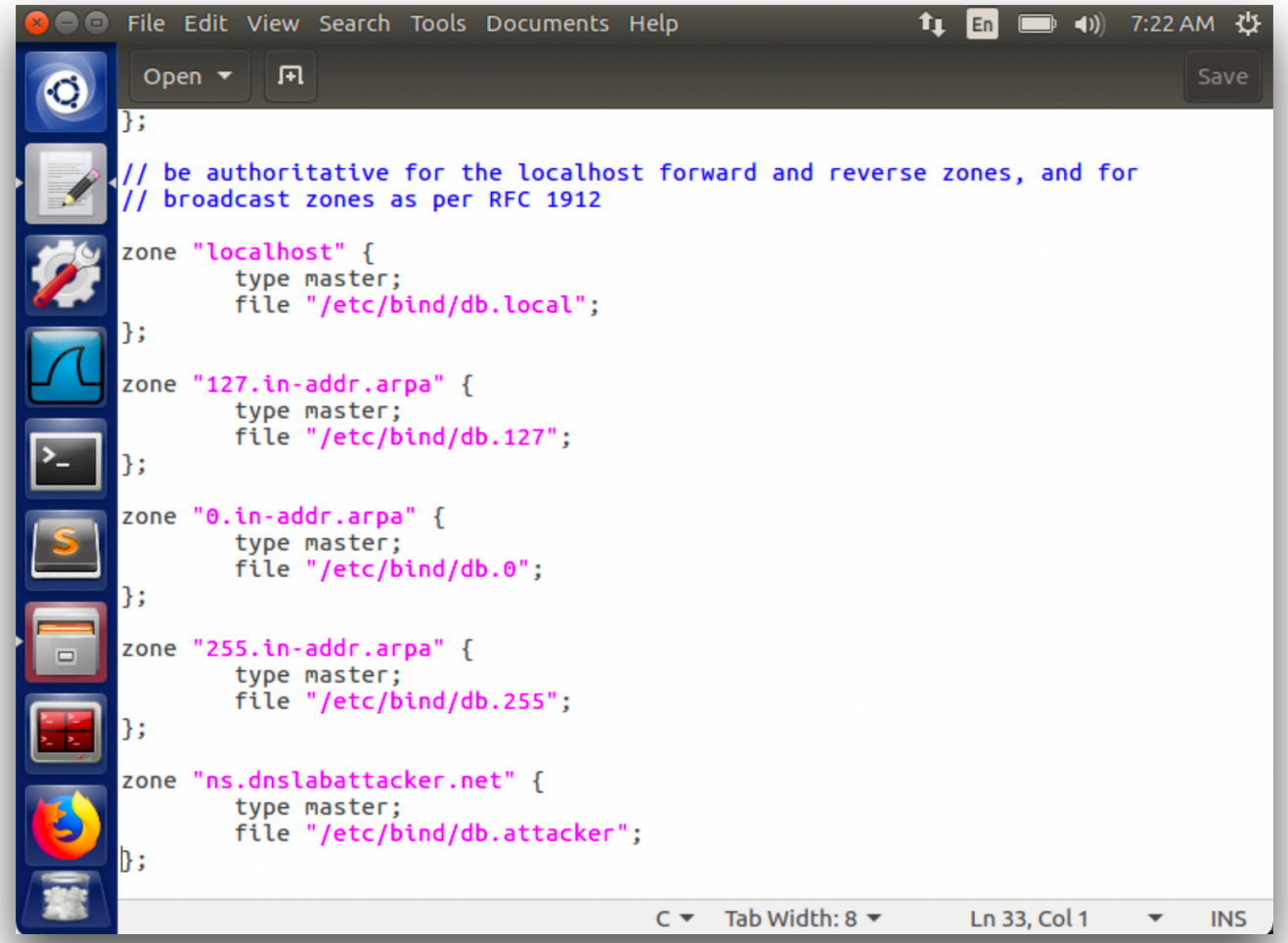
1. We first configure the victim's DNS server Apollo. We then find the file named.conf.default-zones in the /etc/bind/ folder, and add the following entry to it:

zone "ns.dnslabattacker.net" {

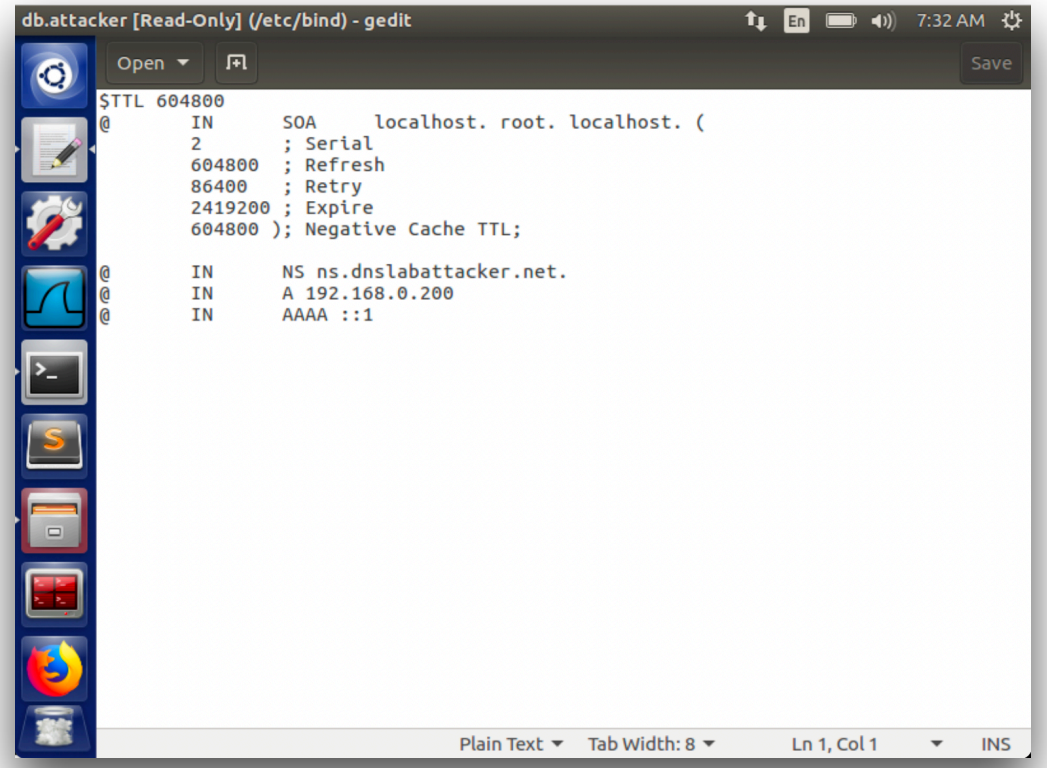
type master;

file "/etc/bind/db.attacker";

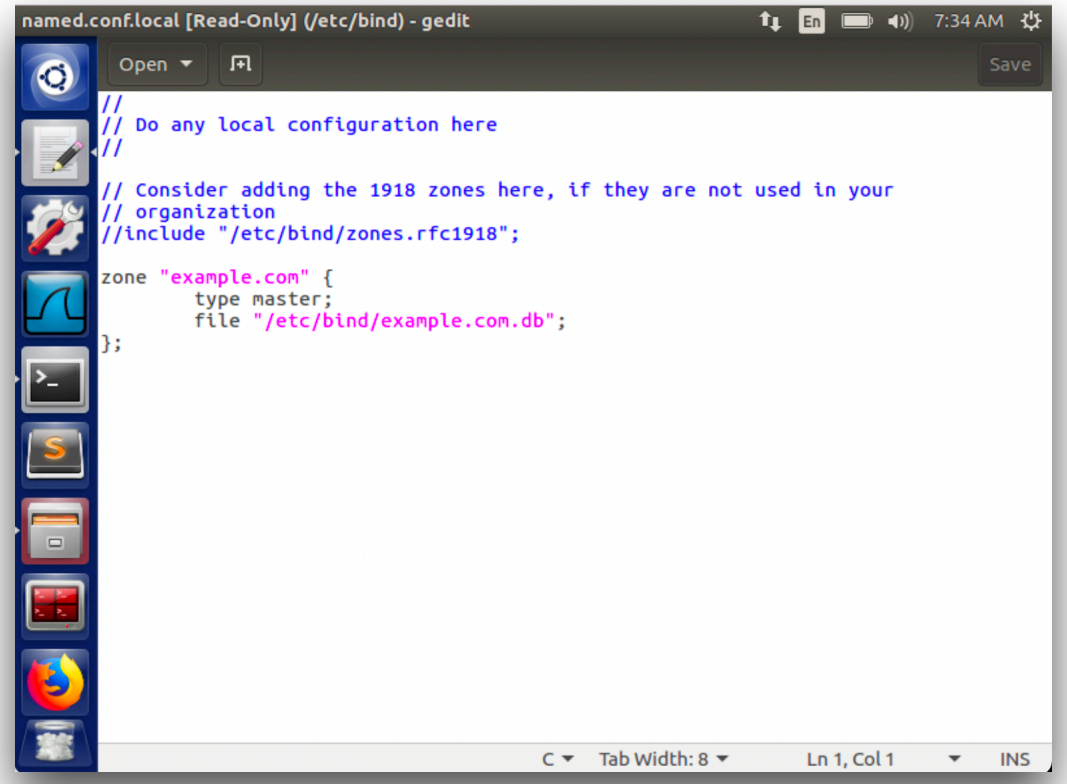
};



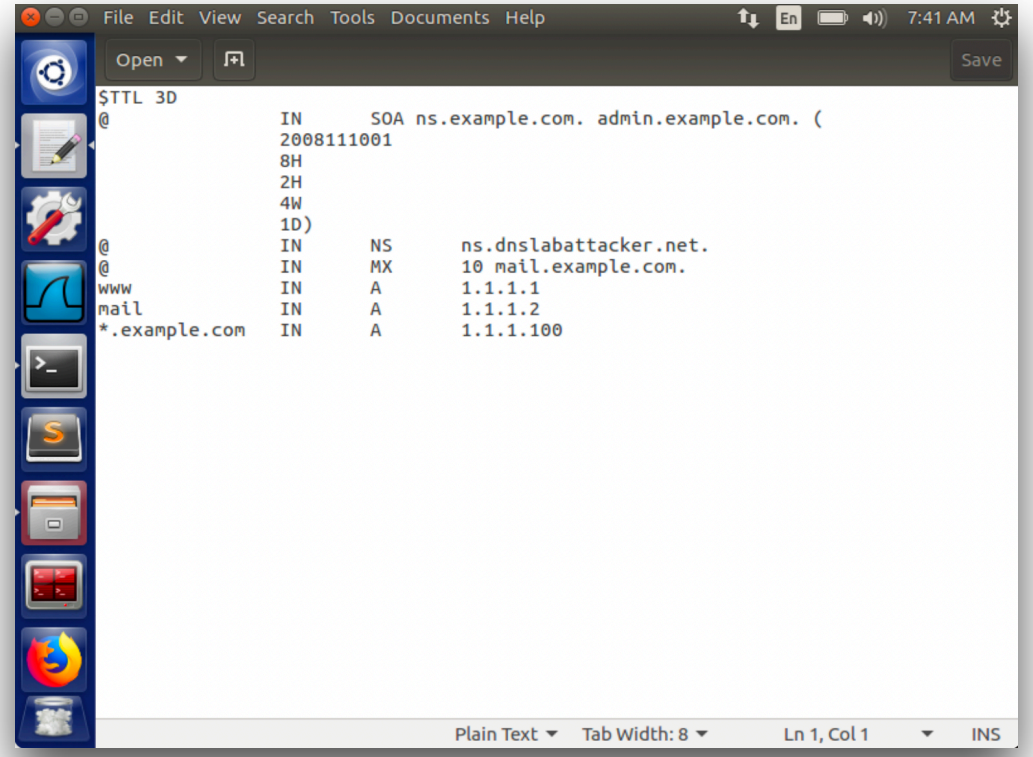
2. We let the attacker's machine and ns.dnslabattacker.net share the machine (192.168.0.200) by creating a file /etc/bind/db.attacker, and placing the following contents in it:



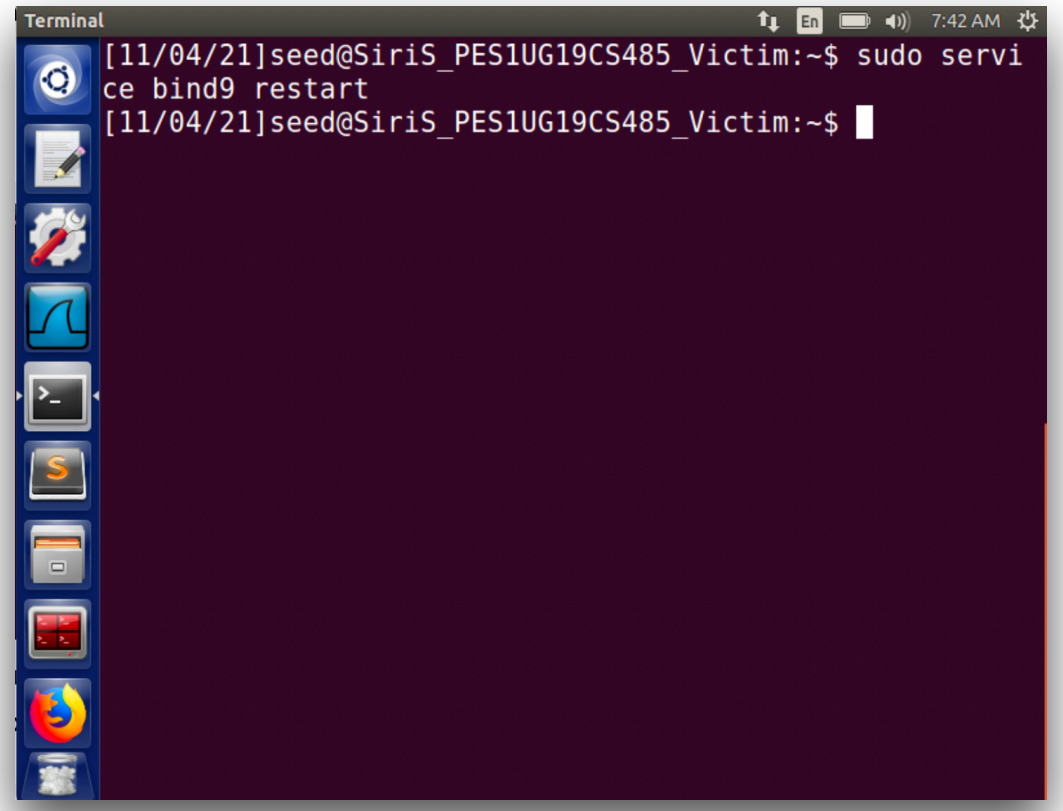
3. We need to configure the DNS server, so it answers the queries for the domain example.com, therefore we add the following entry in /etc/bind/named.conf.local :



4. We then create a file called /etc/bind/example.com.db, and fill it with the following contents.



5. Now we restart the DNS server to make sure all the changes are made:



We observe that the cache has been successful as nameserver shows ns.dnslabattacker.net and the IP address of example.com shows 1.1.1.1