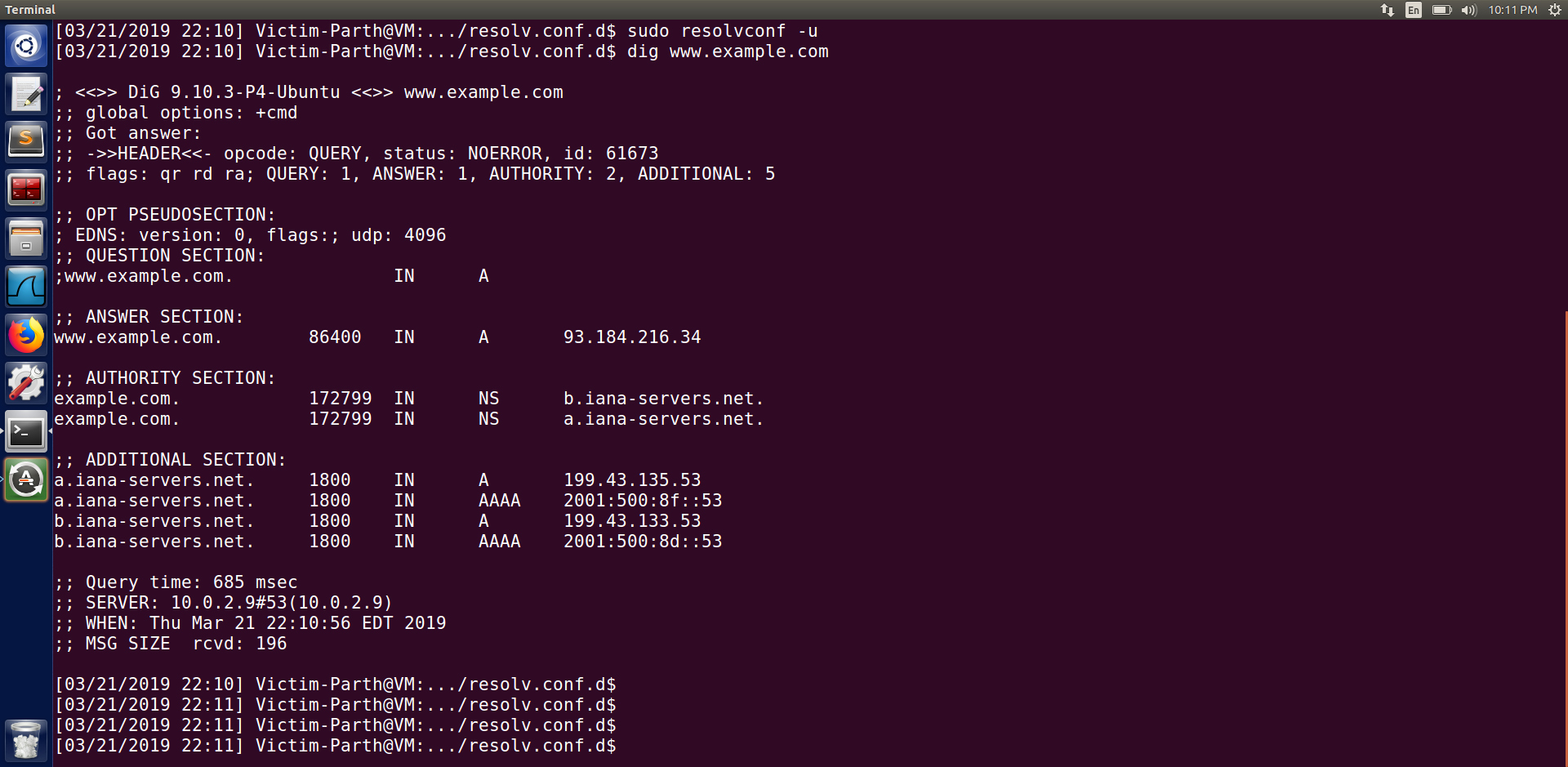
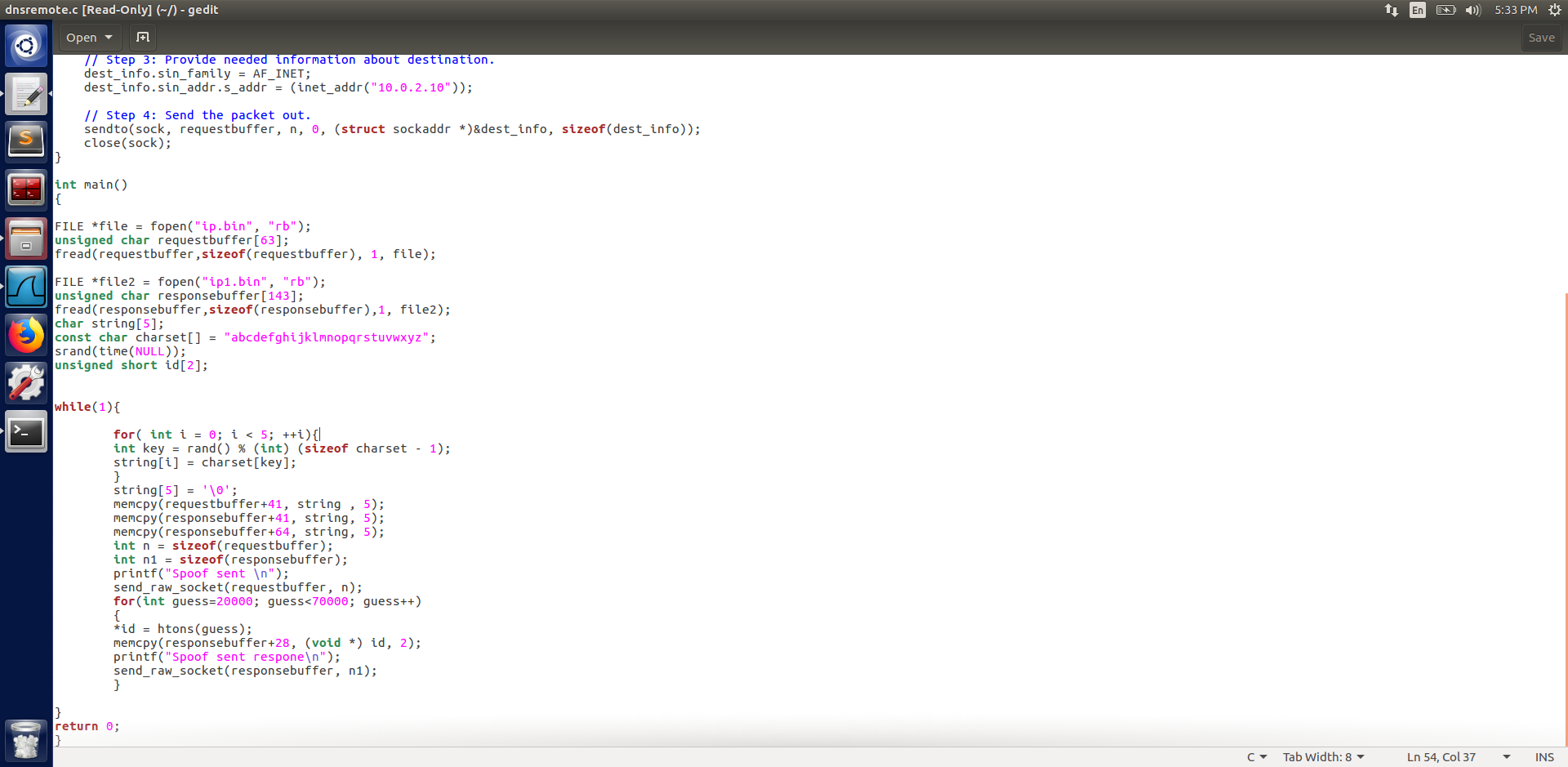
Remote DNS Cache Poisoning Attack Lab

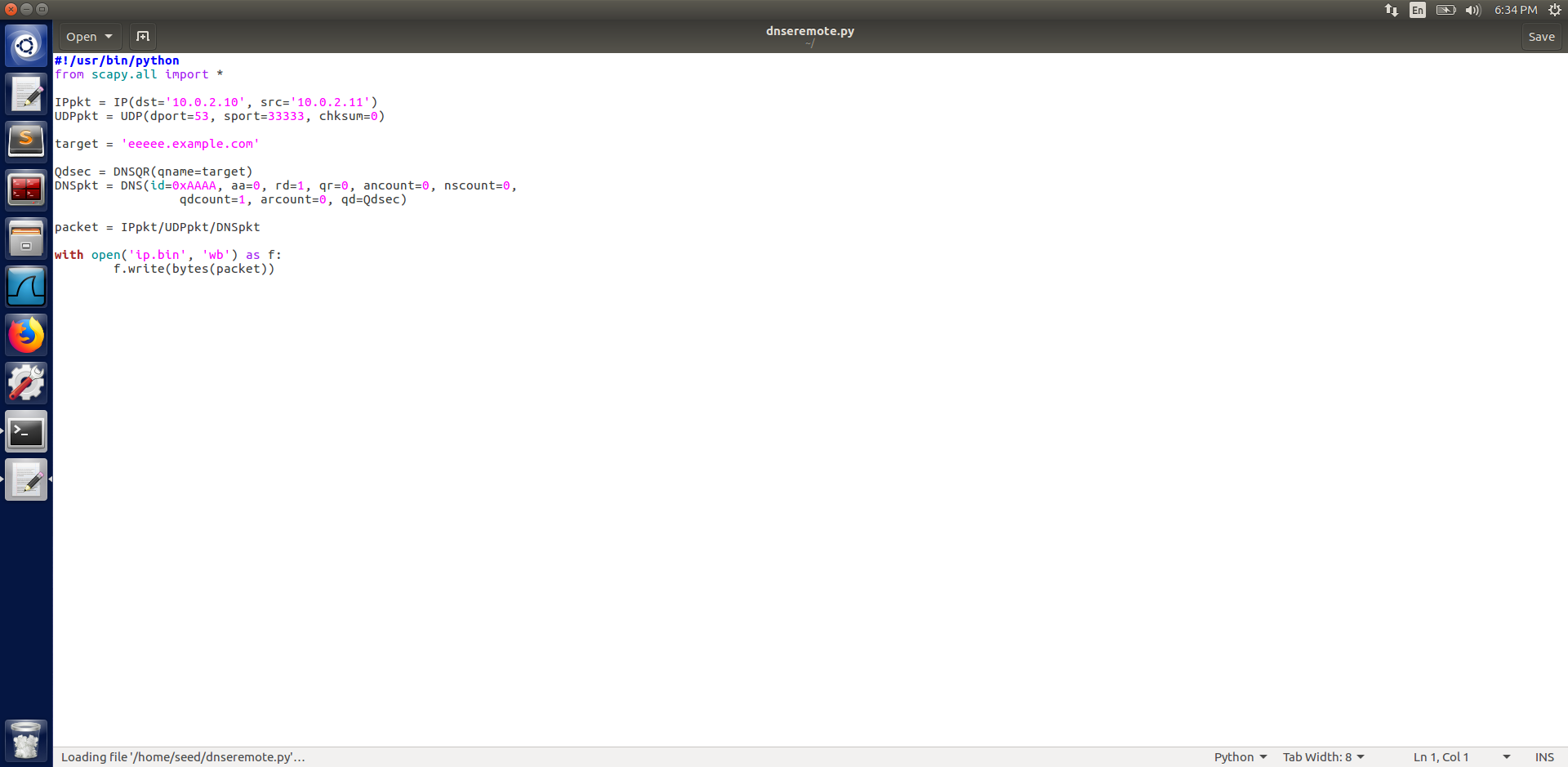


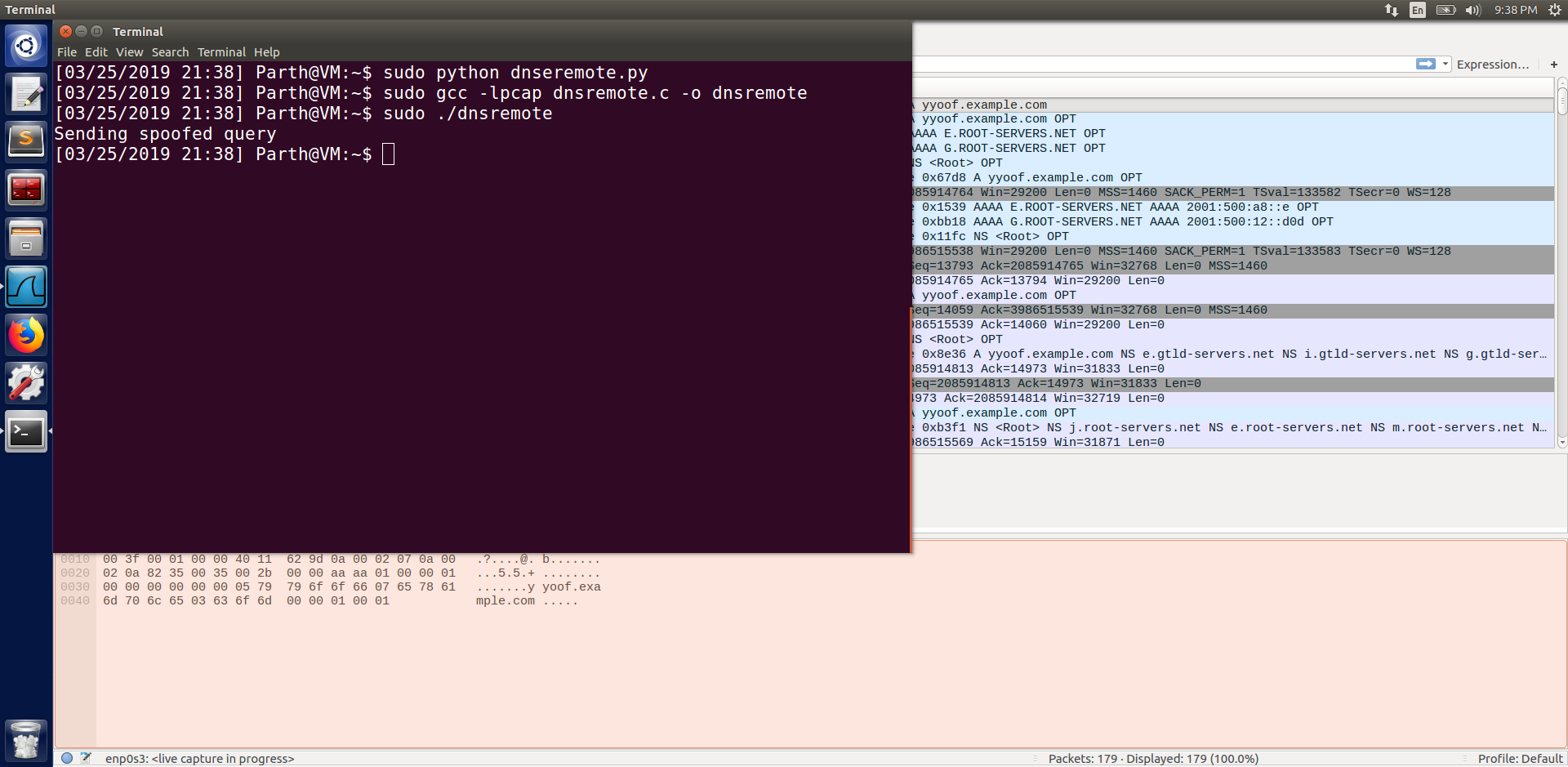
Dnsremote.c Code :

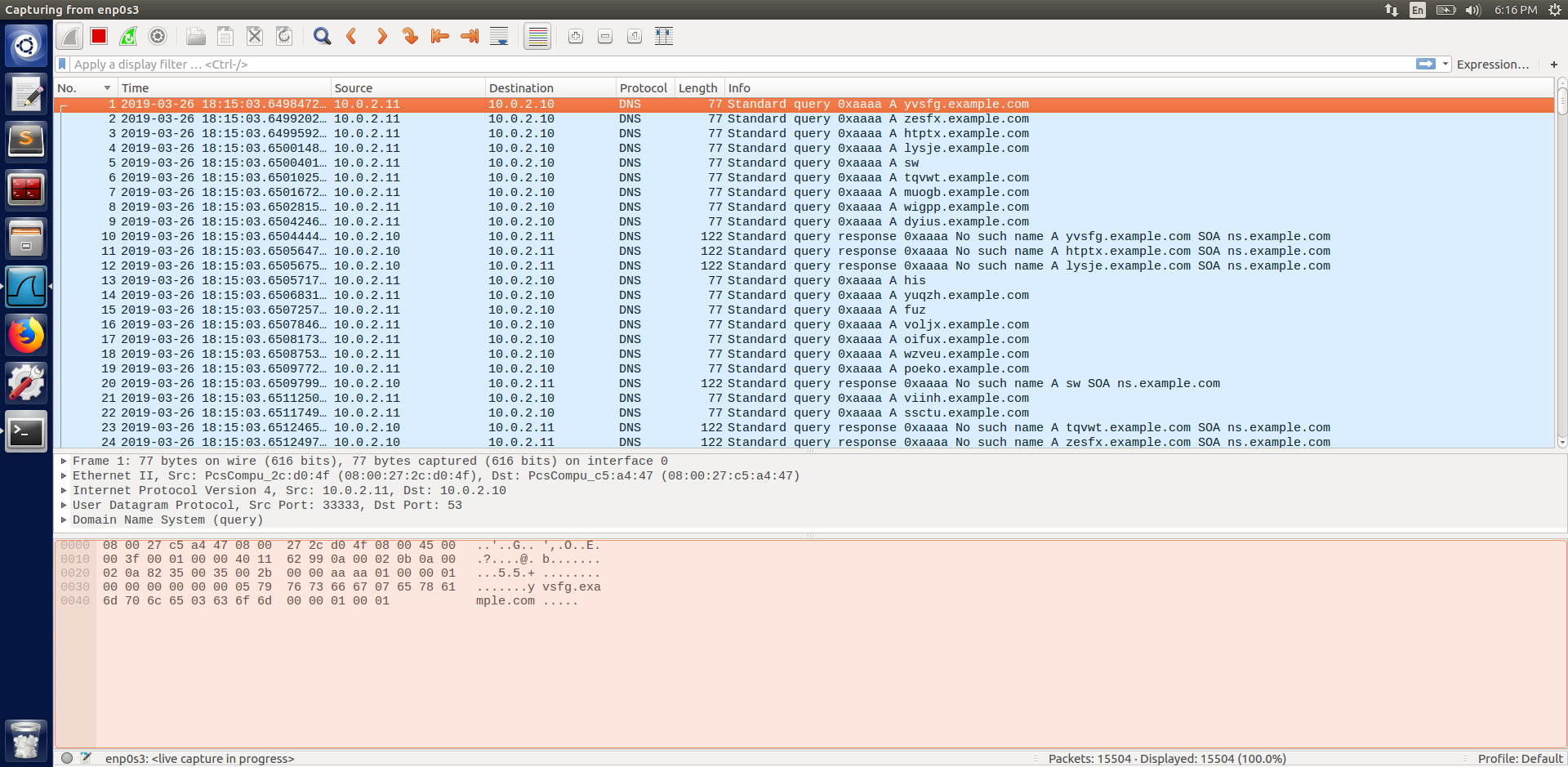




Task 1.1 Spoofing DNS Request

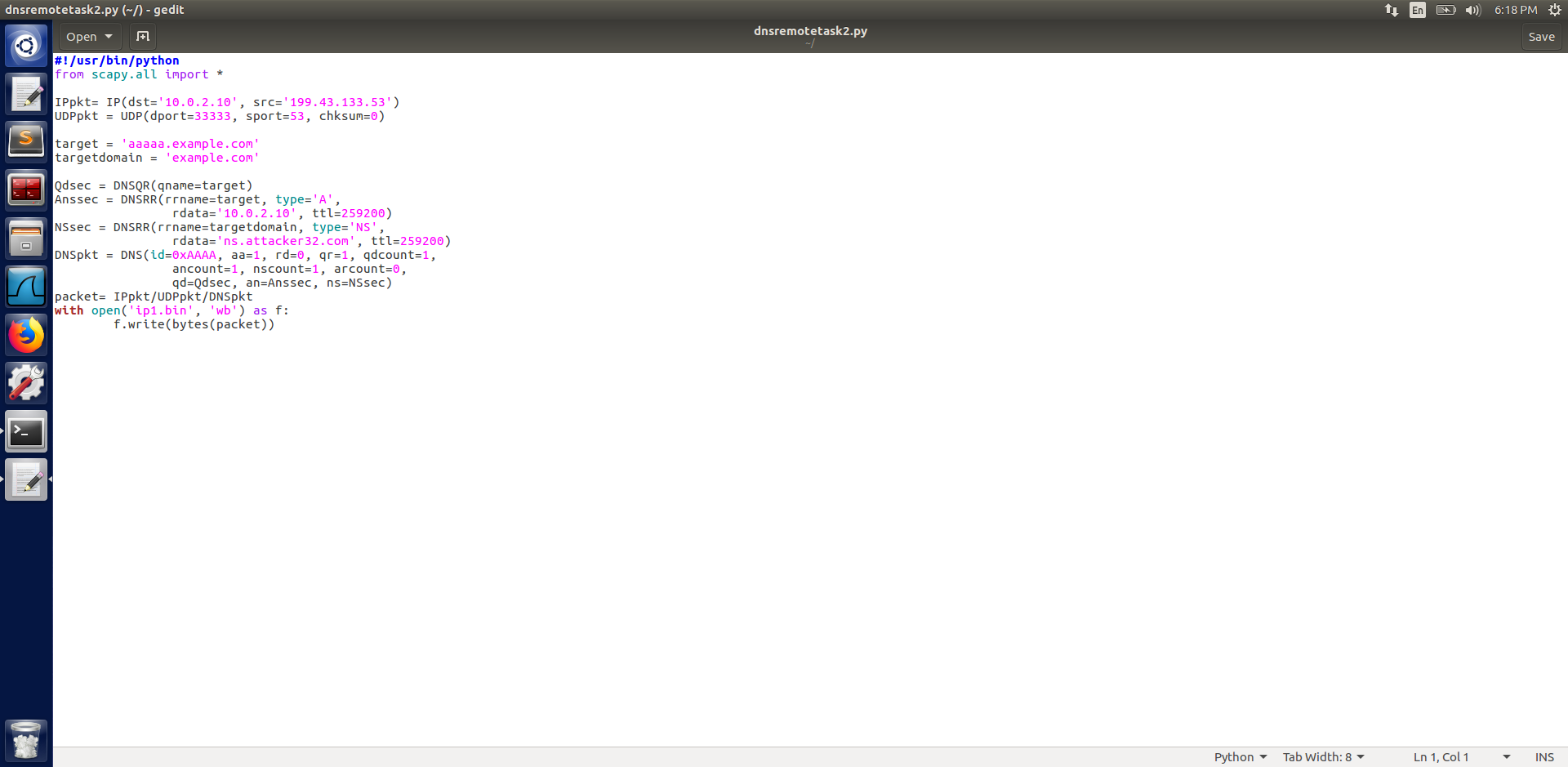


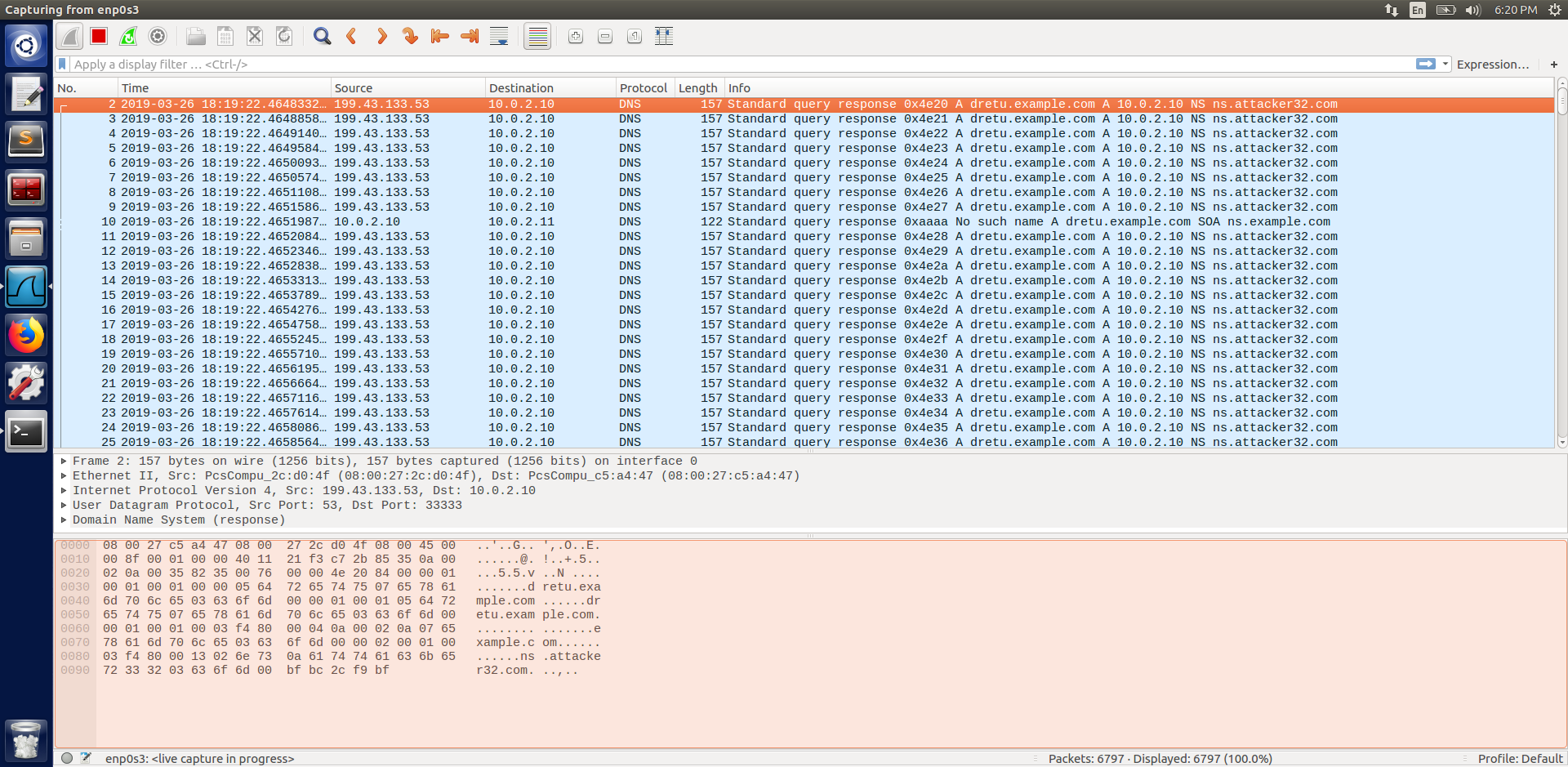




We can see from the code above that we first created the packet in python after which we loaded the file to C and then executed the C code. We can see from the screenshot above that we were successful in spoofing from client to DNS server. After which the DNS server forwarded the packet to example.com

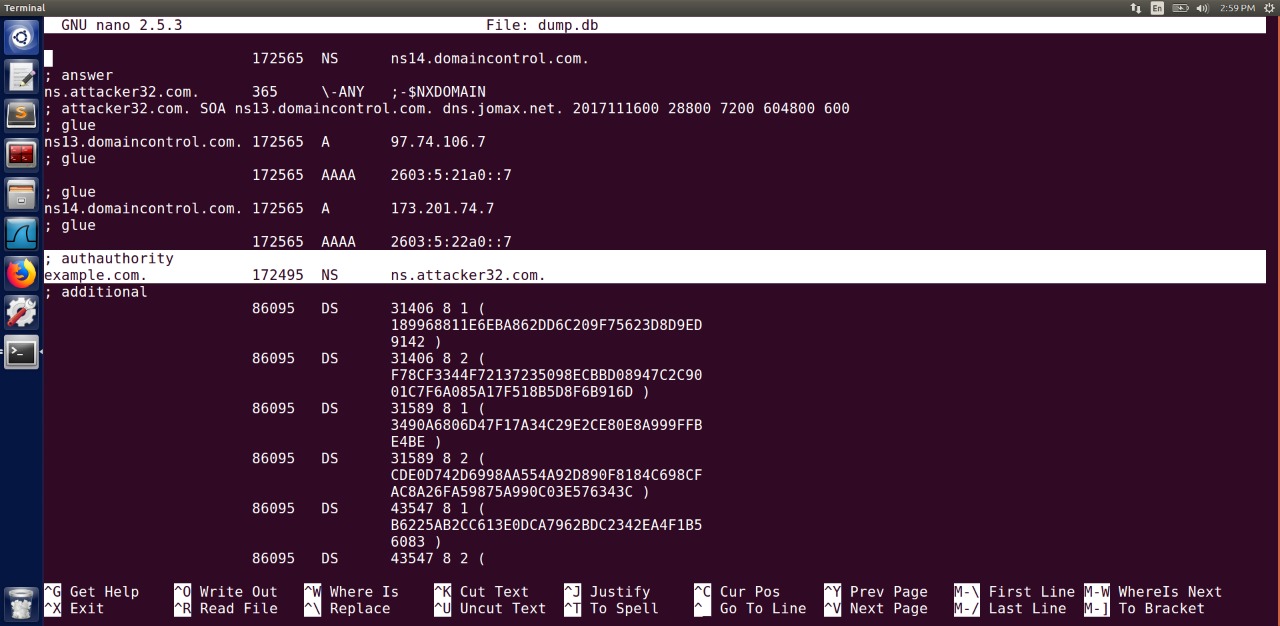
Task 1.2 Spoofing DNS reply





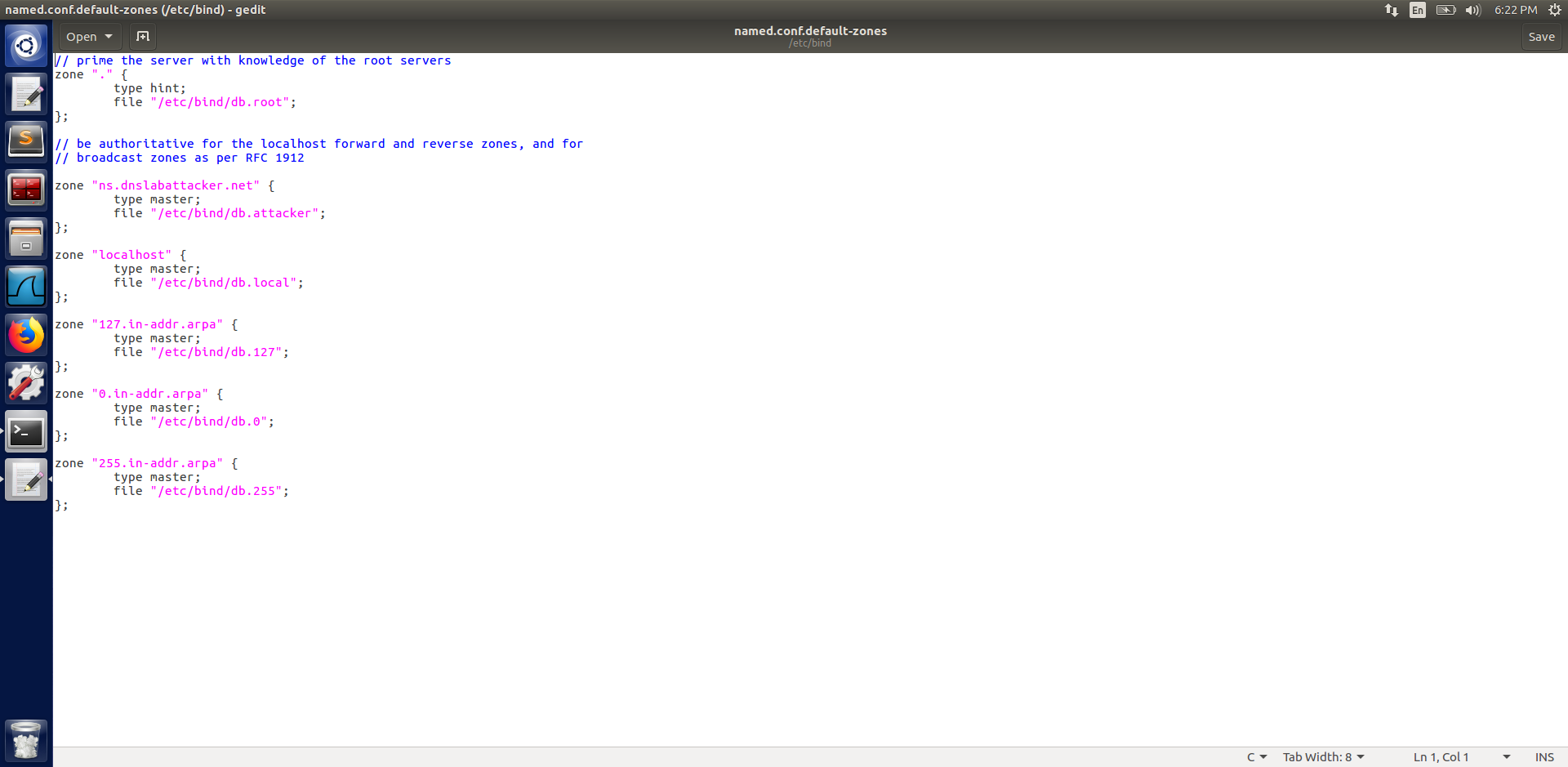
We can see from the above screenshot that we were successful in spoofing the reply. Also we can observe that the subdomain of example.com is randomly generated.

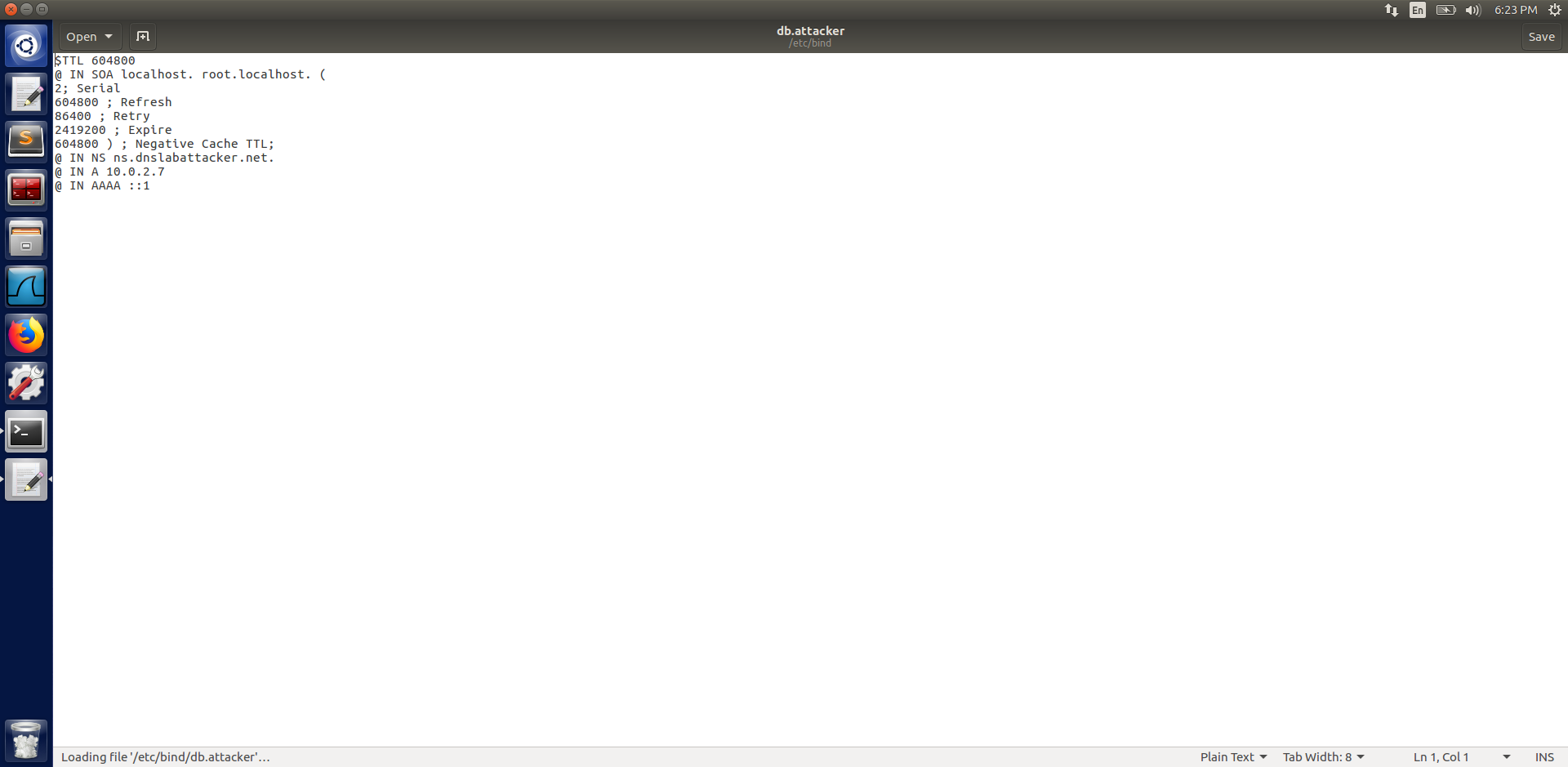
Task 1.3 The Kaminsky Attack

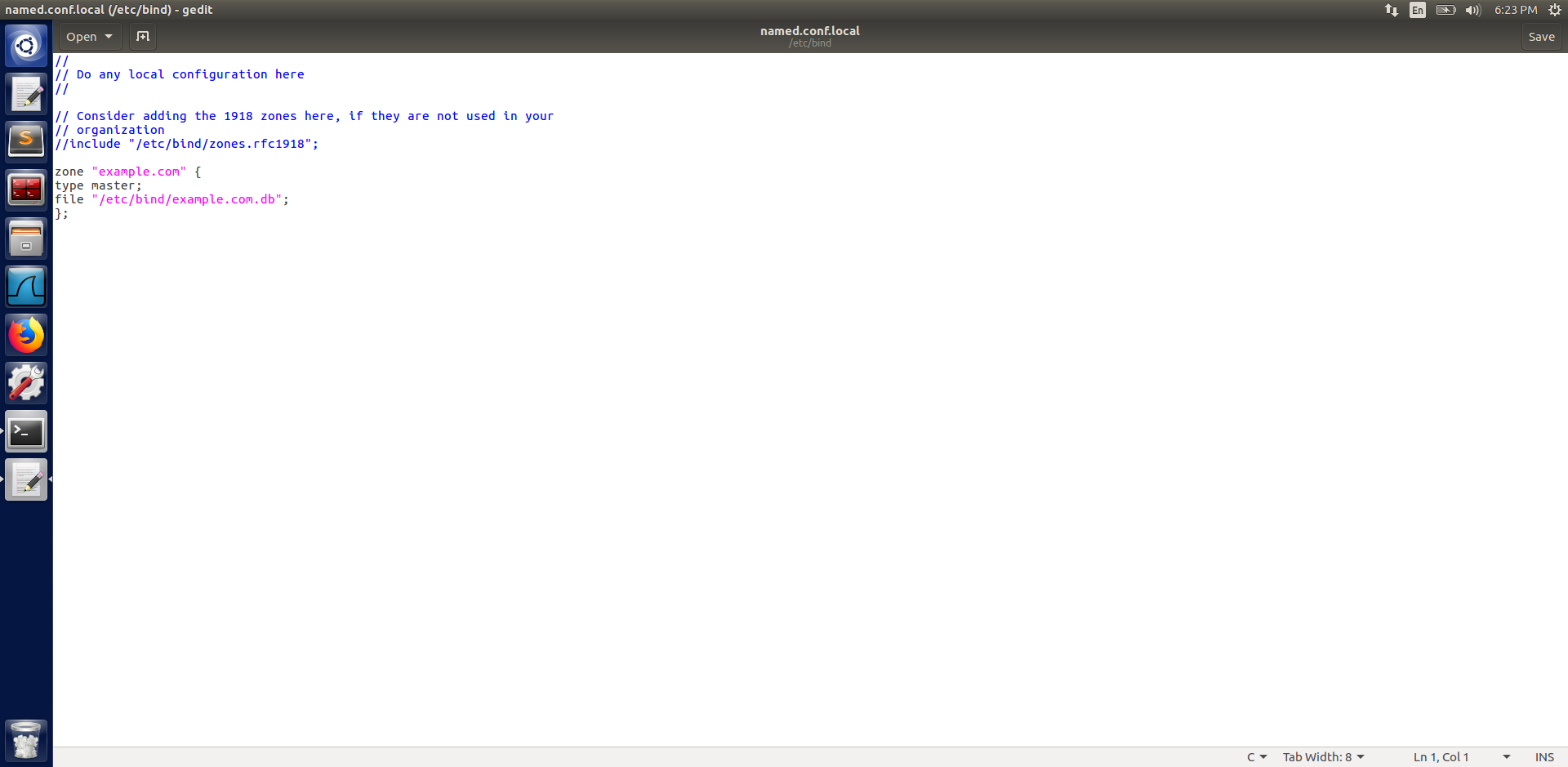


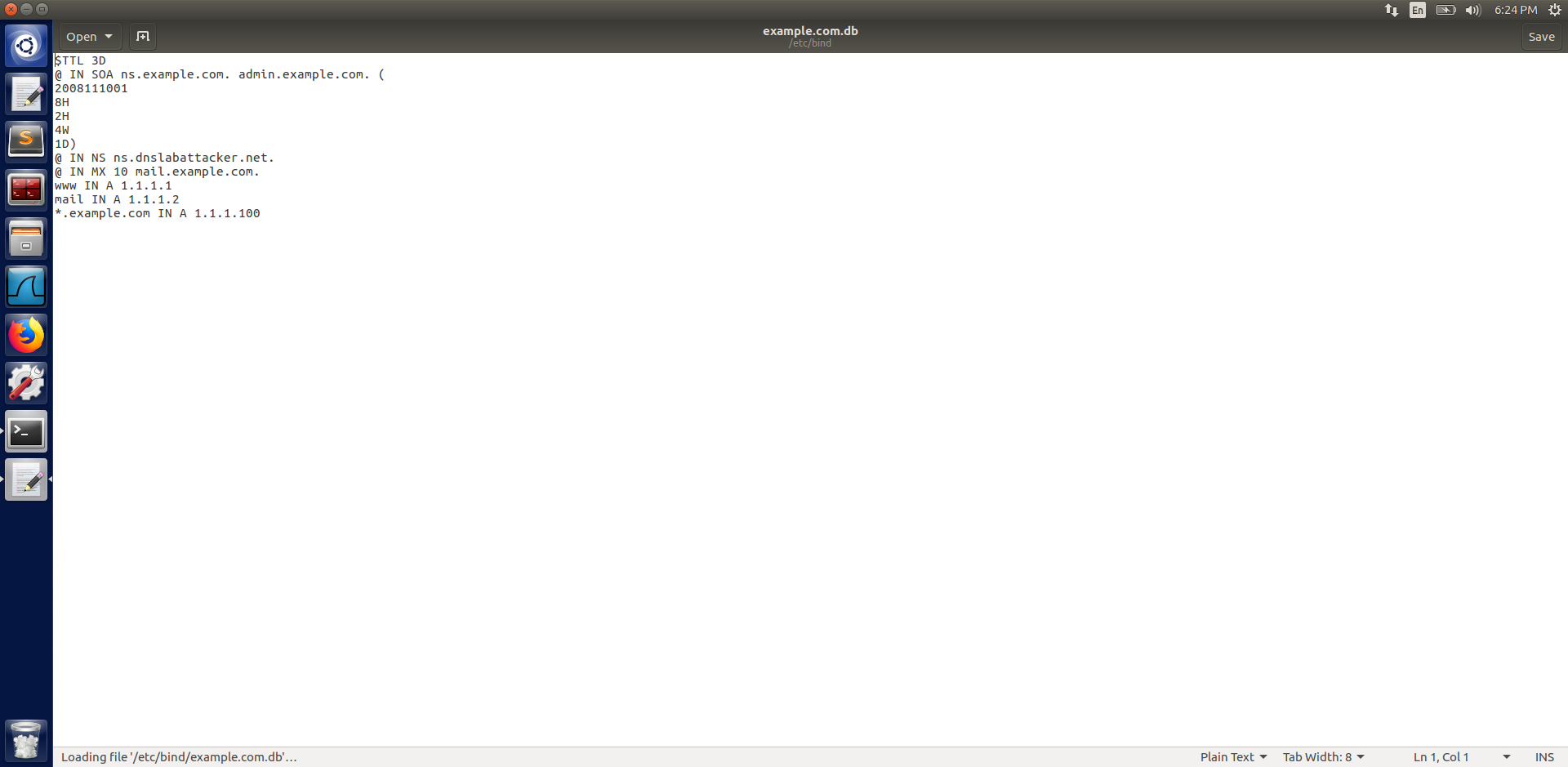
In this task we merge the task 1.1 and 1.2. We can see that the nameserver is the original nameserver as we haven’t yet configured the zone file.

Task 2

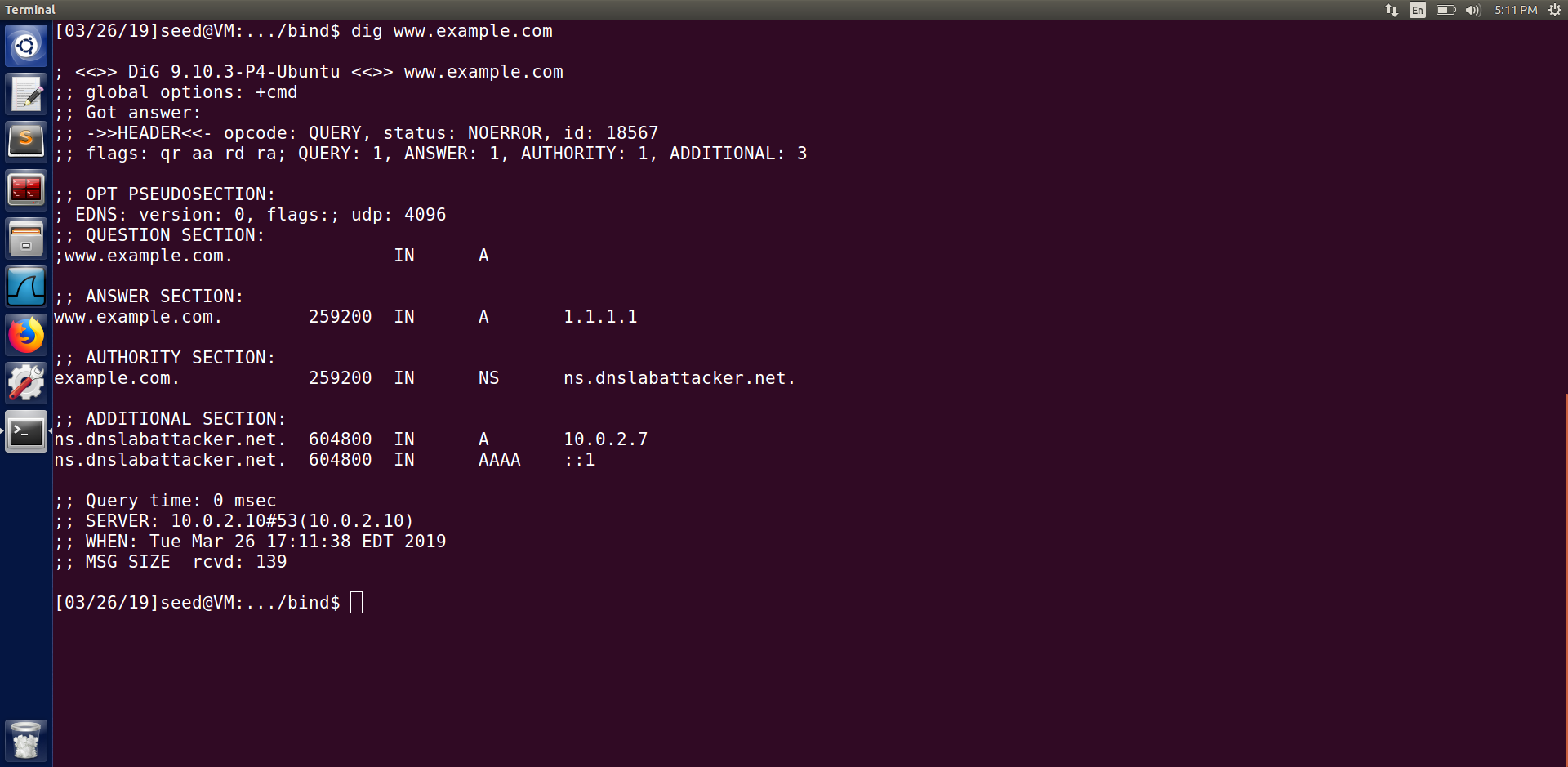








All of the above screenshot under task 2 are for the setup to poison the cache since we are using a fake domain name.



In this final task the victim dns server was successfully poisoned with the nameserver as ns.dnslabattacker.net as seen from the screenshot above.

Why the IP address for ns.dnslabattacker.net in the additional field is not accepted by the victim DNS server.?

Information in additional section in the packet is even though cached by the victim DNS server, it does not trust this information and fetches the information on its own by sending out the query.