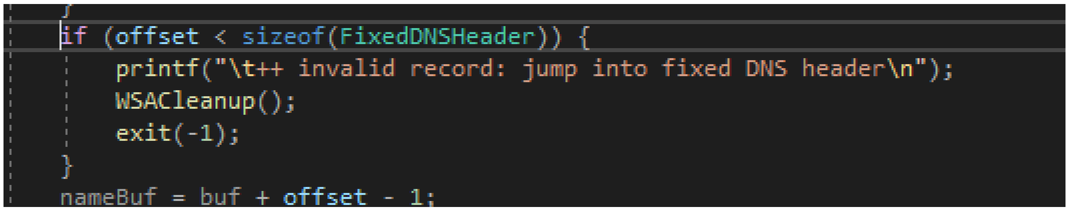
Jared Jones
CSCE 612
Hw 2 Report

Forensic Investigation on Custom DNS Server

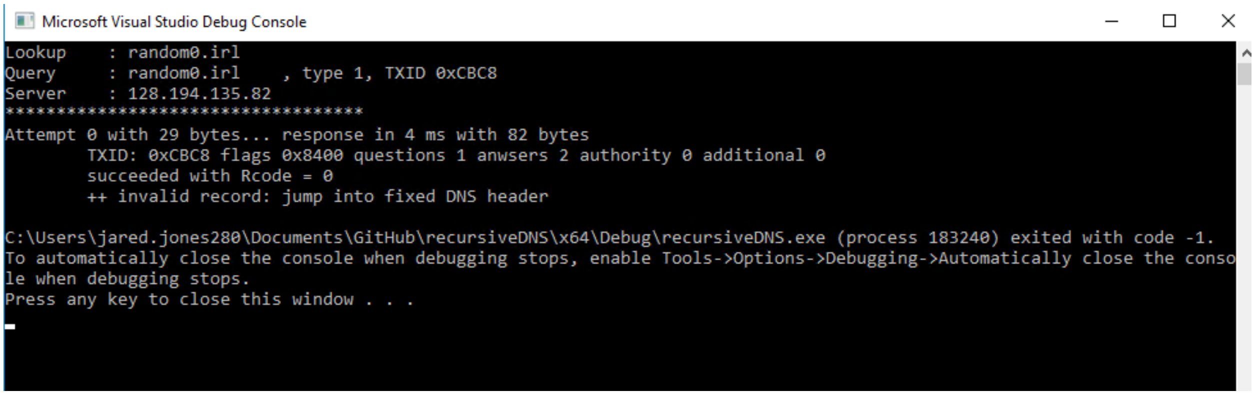
Using experimentation and analysis, determine what types of corruption is performed in each of the cases below and show the corresponding traces from your program with the ++ error it detected.

1. (8 pts) Case 1: random0.irl, random3.irl, random5.irl, and random6.irl.

Random0.irl - Jumps into the fixed dns header, when it jumps the offset is smaller than the size of the fixed dns header at the beginning of the response which means it is jumping into the fixed header. My check for this and trace are below.

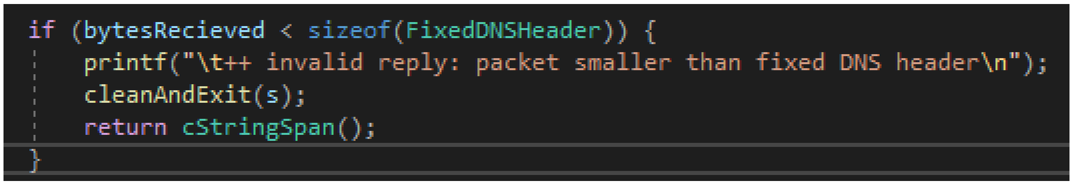


```
    }
    if (offset < sizeof(FixedDNSHeader)) {
        printf("\t++ invalid record: jump into fixed DNS header\n");
        WSACleanup();
        exit(-1);
    }
    nameBuf = buf + offset - 1;
```

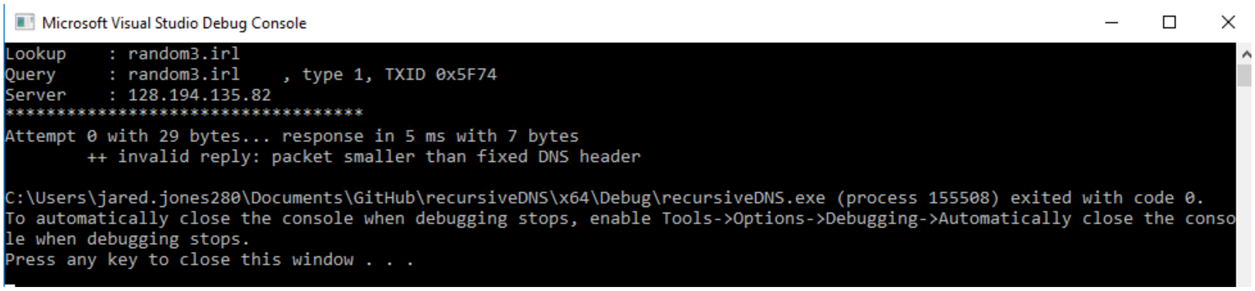


```
Lookup    : random0.irl
Query     : random0.irl    , type 1, TXID 0xCBC8
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 4 ms with 82 bytes
        TXID: 0xCBC8 flags 0x8400 questions 1 anwsers 2 authority 0 additional 0
        succeeded with Rcode = 0
        ++ invalid record: jump into fixed DNS header

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 183240) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Random3.irl - Replies with a packet smaller than the fixed dns header therefore the response is an invalid packet. My check and trace are below.

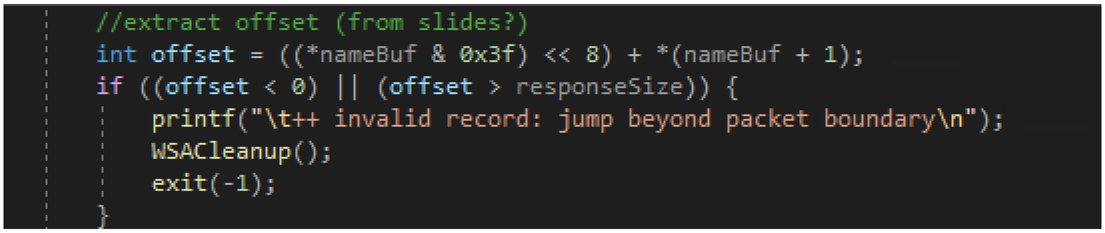


```
if (bytesRecieved < sizeof(FixedDNSHeader)) {
    printf("\t++ invalid reply: packet smaller than fixed DNS header\n");
    cleanAndExit(s);
    return cStringSpan();
}
```

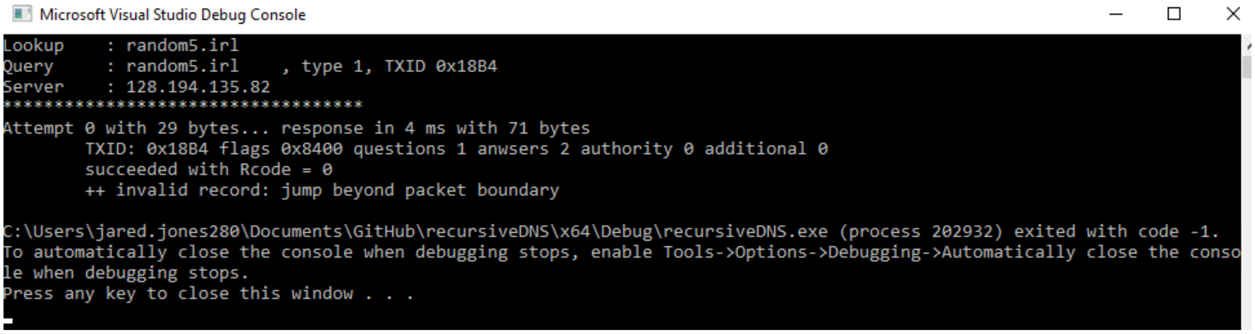


```
Lookup    : random3.irl
Query     : random3.irl    , type 1, TXID 0x5F74
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 5 ms with 7 bytes
        ++ invalid reply: packet smaller than fixed DNS header

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 155508) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Random5.irl - Jumps beyond the packet boundary size. When doing jumps, I keep track of the size of the packet I received and if the jump offset is greater than that packet size, then it triggers this error and stops reading.



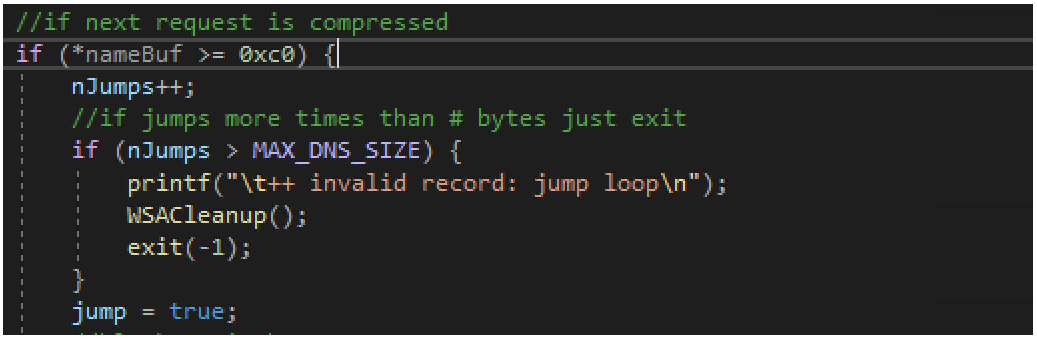
```
    //extract offset (from slides?)
    int offset = ((*nameBuf & 0x3f) << 8) + *(nameBuf + 1);
    if ((offset < 0) || (offset > responseSize)) {
        printf("\t++ invalid record: jump beyond packet boundary\n");
        WSACleanup();
        exit(-1);
    }
```



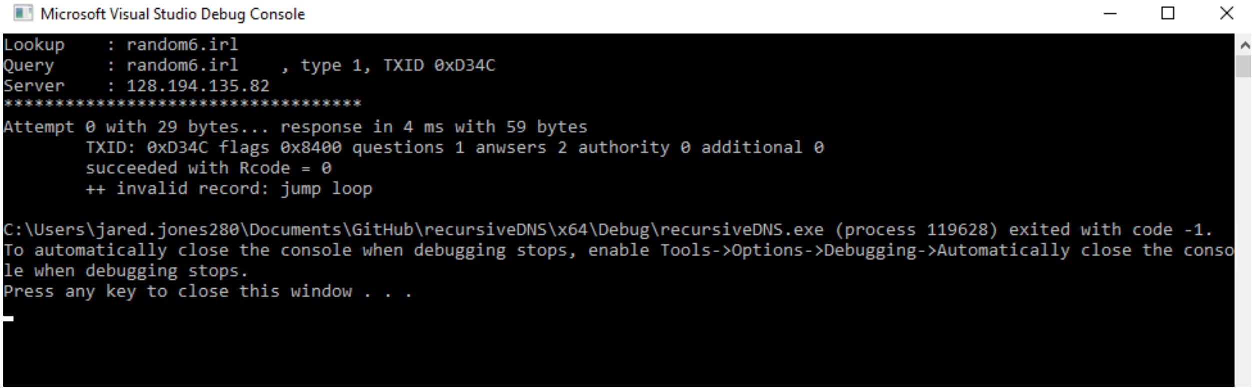
Microsoft Visual Studio Debug Console

```
Lookup    : random5.irl
Query     : random5.irl    , type 1, TXID 0x18B4
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 4 ms with 71 bytes
        TXID: 0x18B4 flags 0x8400 questions 1 anwsers 2 authority 0 additional 0
        succeeded with Rcode = 0
        ++ invalid record: jump beyond packet boundary

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 202932) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Random6.irl - this case returns a response with a jump loop. This is tracked by keeping track of the number of jumps done per name that is being read and then if that number gets too high it exits.



```
//if next request is compressed
if (*nameBuf >= 0xc0) {
    nJumps++;
    //if jumps more times than # bytes just exit
    if (nJumps > MAX_DNS_SIZE) {
        printf("\t++ invalid record: jump loop\n");
        WSACleanup();
        exit(-1);
    }
    jump = true;
```



Microsoft Visual Studio Debug Console

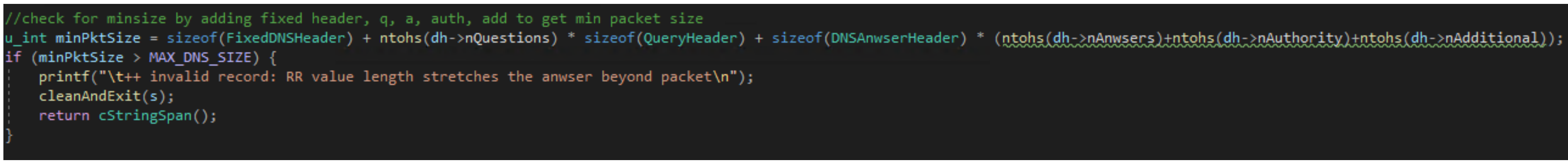
```
Lookup    : random6.irl
Query     : random6.irl    , type 1, TXID 0xD34C
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 4 ms with 59 bytes
        TXID: 0xD34C flags 0x8400 questions 1 anwsers 2 authority 0 additional 0
        succeeded with Rcode = 0
        ++ invalid record: jump loop

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 119628) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

2. (2 pts) Case 2: **random1.irl**.

This case has a record whos RR value length stretches the answer beyond the range of the max value dns allows to be transmitted. It returns 1 question, 1 answer, 0 authorities, and

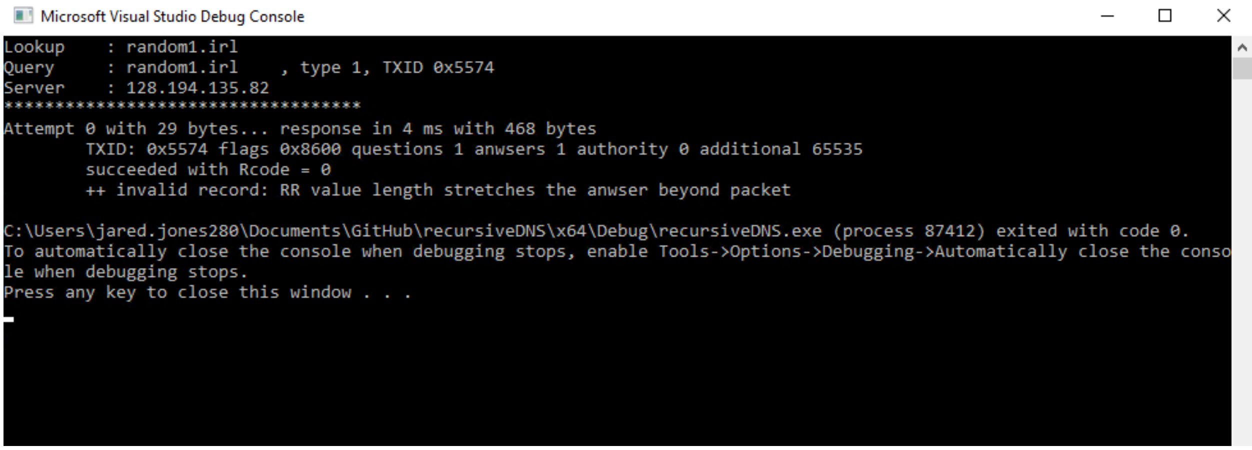
65535 additional which just the size of the headers alone for those answers would be larger than the max dns packet size of 512 bytes.

My check:



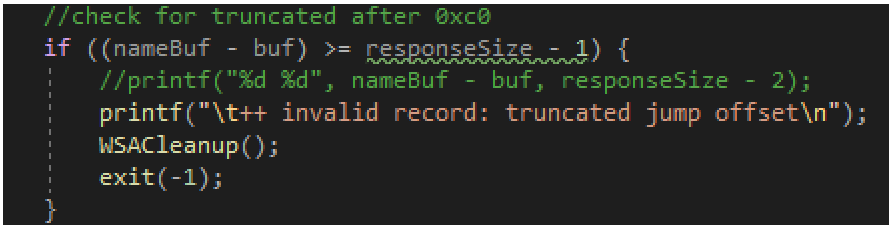
```
//check for minsize by adding fixed header, q, a, auth, add to get min packet size
u_int minPktSize = sizeof(FixedDNSHeader) + ntohs(dh->nQuestions) * sizeof(QueryHeader) + sizeof(DNSAnwserHeader) * (ntohs(dh->nAnwsers)+ntohs(dh->nAuthority)+ntohs(dh->nAdditional));
if (minPktSize > MAX_DNS_SIZE) {
    printf("\t++ invalid record: RR value length stretches the anwser beyond packet\n");
    cleanAndExit(s);
    return cStringSpan();
}
```

My trace:

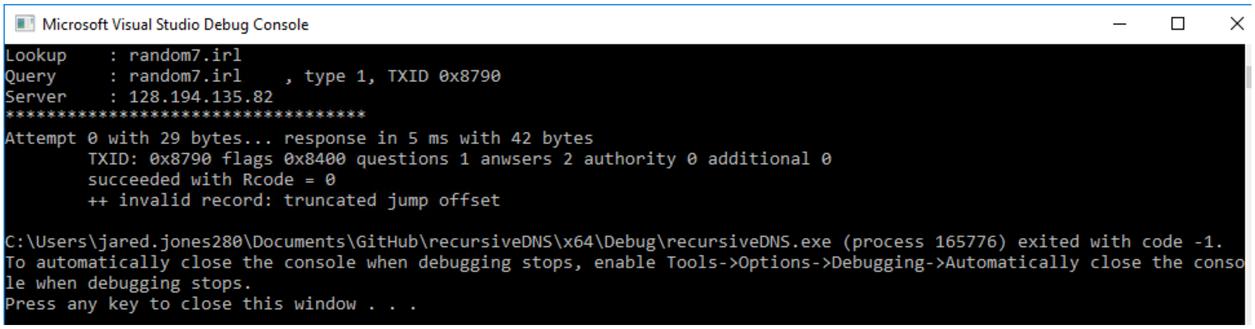


```
Lookup    : random1.irl
Query     : random1.irl    , type 1, TXID 0x5574
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 4 ms with 468 bytes
        TXID: 0x5574 flags 0x8600 questions 1 anwsers 1 authority 0 additional 65535
        succeeded with Rcode = 0
        ++ invalid record: RR value length stretches the anwser beyond packet

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 87412) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

3. (3 pts) Case 3: **random7.irl** - has a truncated jump offset at the end of the packet that would cause the jump offset to be calculated from external memory that is not where the server is storing the response. This is checked when a jump is found, its location is checked to make sure there minimum 1 byte of space in the rest of the packet to ensure that the offset will be able to be calculated correctly.



```
//check for truncated after 0xc0
if ((nameBuf - buf) >= responseSize - 1) {
    //printf("%d %d", nameBuf - buf, responseSize - 2);
    printf("\t++ invalid record: truncated jump offset\n");
    WSACleanup();
    exit(-1);
}
```



```
Lookup    : random7.irl
Query     : random7.irl    , type 1, TXID 0x8790
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 5 ms with 42 bytes
        TXID: 0x8790 flags 0x8400 questions 1 anwsers 2 authority 0 additional 0
        succeeded with Rcode = 0
        ++ invalid record: truncated jump offset

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 165776) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

4. (12 pts) Case 4: **random4.irl**.

Show three types of ++ errors produced by this query that are not present in any of the cases above and document your handling of each. Since these responses are randomized, you will need to run your program multiple times. The cases above should cover all nine ++ errors stated earlier.

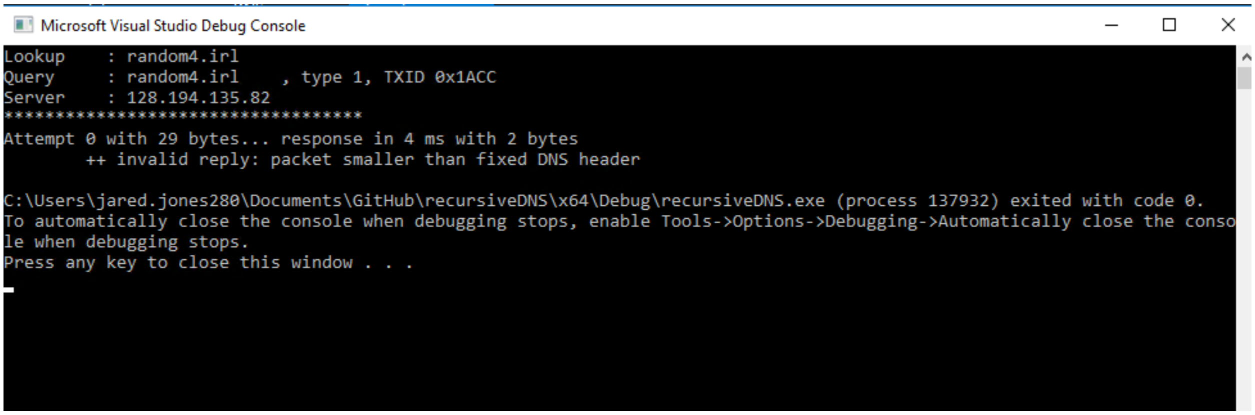
I found 4 different responses from random4.irl and 3 of them were all different from the first errors I ran, making them cover all 9 errors stated earlier.

They were:

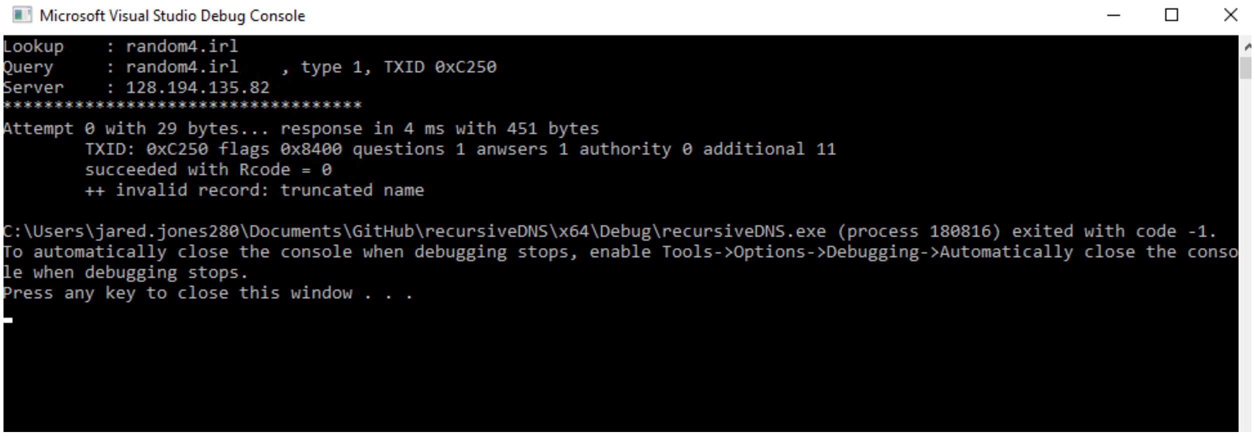
Packet smaller than fixed dns header
Truncated name
Truncated RR answer header
Not enough records

I have put my traces below. For the three errors that were unique to random4.irl, they were all able to be checked for by comparing the location of my current char* pointer subtracted from the beginning of the packet (to get the current byte in the packet we were on) against the maximum number of bytes received in the message. Depending on where you were in the parsing of the record, name, or header determined exactly which error you got. If you were at the beginning of parsing a new record and found you were at the end of the packet, you got “not enough records”, if you were in the middle of parsing a name you got “truncated name” and if you were casting the header onto the char array and found that adding the header size to the current pointer took you over then you got “truncated RR answer header”.

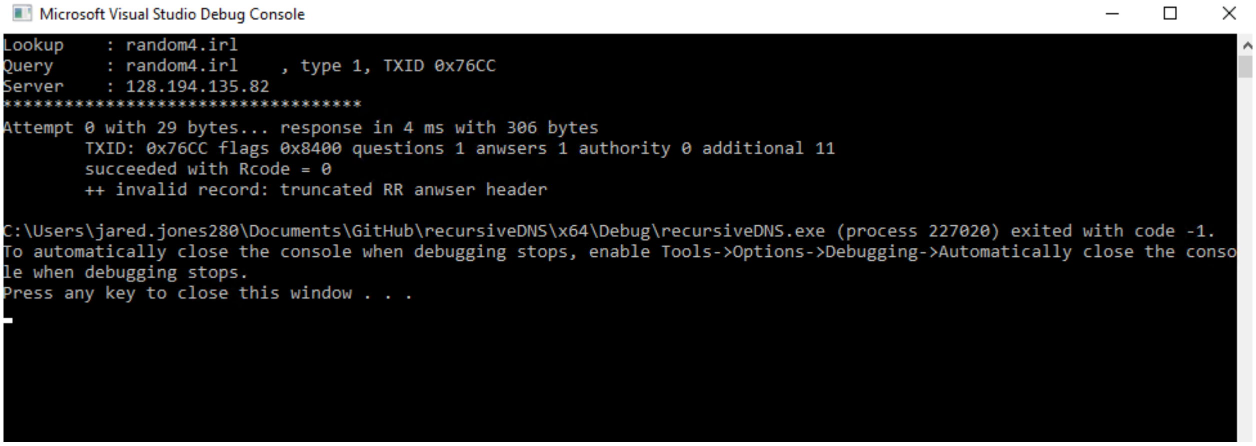
This was an example of 3 errors that you can check for with the same comparison, just in different locations. My suspicion is that my 2 byte response from random4.irl is just a quirk response, because I only got it once while the others all were repeatable.



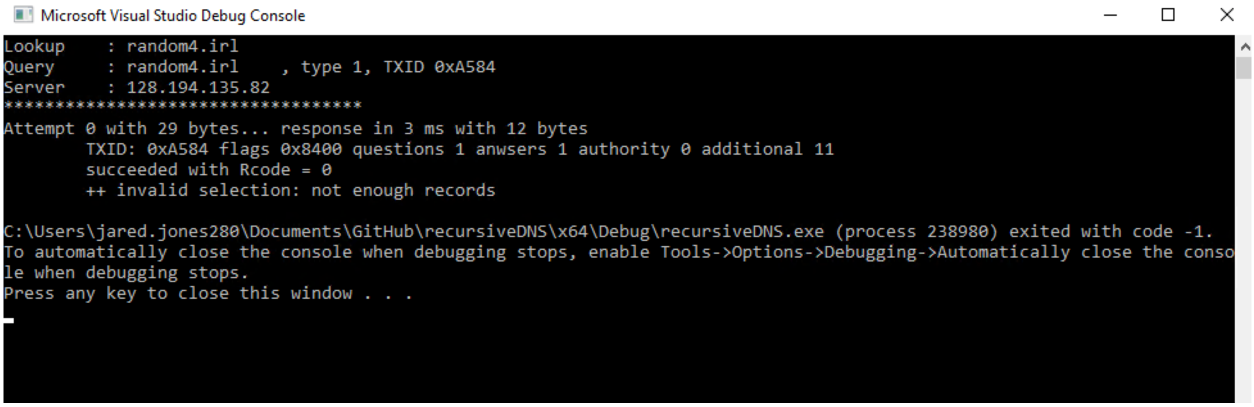
```
Lookup    : random4.irl
Query     : random4.irl    , type 1, TXID 0x1ACC
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 4 ms with 2 bytes
        ++ invalid reply: packet smaller than fixed DNS header

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 137932) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```



```
Lookup    : random4.irl
Query     : random4.irl    , type 1, TXID 0xC250
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 4 ms with 451 bytes
        TXID: 0xC250 flags 0x8400 questions 1 anwsers 1 authority 0 additional 11
        succeeded with Rcode = 0
        ++ invalid record: truncated name

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 180816) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```



```
Lookup    : random4.irl
Query     : random4.irl    , type 1, TXID 0x76CC
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 4 ms with 306 bytes
        TXID: 0x76CC flags 0x8400 questions 1 anwsers 1 authority 0 additional 11
        succeeded with Rcode = 0
        ++ invalid record: truncated RR anwser header

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 227020) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```



```
Lookup    : random4.irl
Query     : random4.irl    , type 1, TXID 0xA584
Server    : 128.194.135.82
***********************************
Attempt 0 with 29 bytes... response in 3 ms with 12 bytes
        TXID: 0xA584 flags 0x8400 questions 1 anwsers 1 authority 0 additional 11
        succeeded with Rcode = 0
        ++ invalid selection: not enough records

C:\Users\jared.jones280\Documents\GitHub\recursiveDNS\x64\Debug\recursiveDNS.exe (process 238980) exited with code -1.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

5. (extra credit, 10 pts): Figure out the algorithm behind **random8.irl's** response. This query generates randomized replies, so you will need to run it several times to see what happens. It is not enough (or even necessary) to report the errors your code detects; instead, you should explain the essence of what the server is doing to the packet so that someone else can write code to implement something similar.

Random8.irl's response is always 1 question, 1 answer, 0 authorities, and 11 additional RR. The Query is the original question it was asked, and the answer is random.irl with an address of 1.1.1.1. It also returns 11 additional records that are the first 11 lines of the scrolling text at the beginning of starwars episode 4 a new hope that total a length of 510 bytes. The message begins to have errors and issues in the additional records section, a random part is replaced with “lol” repeating a random number of times. This can happen at any time in the name, type, class, ttl, data length, or address of the response. This breaks the dns response formatting causing malformed packets.