CSCE 612 / HW2

# Case 1

1. random0.irl

A screenshot of a computer error

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

The previous Wireshark output reveals a jump into the fixed DNS header.

A pointer (0xC0 0x04) directs to an offset within the fixed DNS header. Since 4 falls within the range of 0-12 (the size of the fixed DNS header), this results in an invalid reference.

Thus, the error message correctly identifies the issue.

1. random3.irl

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

The response is only 8 bytes long, which is smaller than the standard 12-byte fixed DNS header. This indicates an issue with the response format.

To further validate this, the Wireshark output below confirms the malformed nature of the packet.

The packet is flagged as a **malformed DNS packet**, **reinforcing** that the response is incorrectly formatted.

1. random5.irl

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

Here we can see that an issue occurred in the second answer, where a jump exceeded the packet boundary of **71 bytes**. Additionally, examining the Wireshark output confirms this issue. The response packet contains a **malformed DNS entry**, as indicated by the error message. The jump at **offset 6A (decimal 106)** attempts to reference data beyond the valid bounds of the packet, leading to a **packet parsing failure**.

1. random6.irl

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A jump loop appears to be present in the first answer of the response. The console output confirms this issue by detecting an invalid record due to a repeated offset (offset 43 revisited).

Wireshark further verifies this problem, showing that the CNAME field contains a pointer that loops back to an earlier position. This recursive reference prevents proper resolution and results in an incorrectly formatted DNS packet.

# Case 2 (random1.irl)

A screenshot of a computer

AI-generated content may be incorrect.

In this DNS query response, the program detected an inconsistency in the number of additional records **declared** versus the **actual number found**. The response header indicates 65,535 additional records, an extremely high and likely incorrect value. However, upon parsing, only 11 additional records were present in the response. Which implies a malformed DNS response.

# Case 3 (random7.irl)

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

The response packet for **random7.irl** contains a truncated jump offset, as indicated by the **0xC0** byte at the end of the packet. Normally, a **0xC0XX** compression pointer should be followed by a valid offset within the packet. However, in this case, **0xC0** appears at the very end, meaning there is no second byte to complete the offset. This results in an invalid record because the jump target cannot be determined, confirming the **"truncated jump offset"** error.

# Case 4 (Random Malformed Responses - random4.irl)

1. truncated name

A computer screen shot of a black screen

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

The DNS response contains a truncated name, as indicated by the final part of the record being cut off before the expected end. The response includes a label with a length of 7 (0x07), but the packet ends before the full name can be read. This is evident in the hex dump, where the expected characters are missing or incomplete.

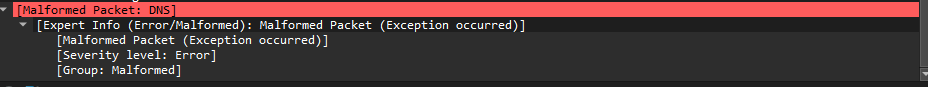
In Wireshark, this is identified as a malformed packet, confirming that the response is incorrectly formatted. The issue occurs because the packet does not contain enough data to complete the name, leading to an error when parsing.

1. RR value length stretches the answer beyond the packet

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

In the highlighted section of the DNS response, we observe the sequence **00 04 02 02 02**. The **04** byte represents the declared length of the RR (Resource Record) value. However, if the remaining packet size is less than the declared **4 bytes**, this results in an out-of-bounds read, leading to the error **"RR value length stretches the answer beyond the packet."** This discrepancy indicates a malformed DNS response where the actual data does not match the specified length, which can be caused by corruption or incorrect encoding of the response.

1. Truncated RR answer header

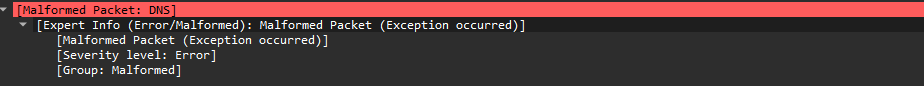
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.





In the response, the additional section includes a **Resource Record (RR) answer header**, which is expected to be **10 bytes** long. However, only **3 bytes** are available before the packet ends. This results in a malformed packet error: **"Truncated RR answer header in Additional (only 3 bytes available, expected 10)"**, as confirmed by both the hex dump and the error message in Wireshark. As highlighted in red, if we compare them we can easily detect how is the second highlighted chunk is truncated.

# Extra Credit

the random8.irl server randomly generates replies that fits a random “lol’s” inside the dns responses. This could lead to several malformed packets which for example includes too long label (label > 63 as per RFC 1035) (as shown below, after the code). Also, some malformed **DNS answer header** responses with different **DNS answer header** Types, Classes, TTLs and Data length that exceeds remaining bytes in the packet (as described in previous cases).

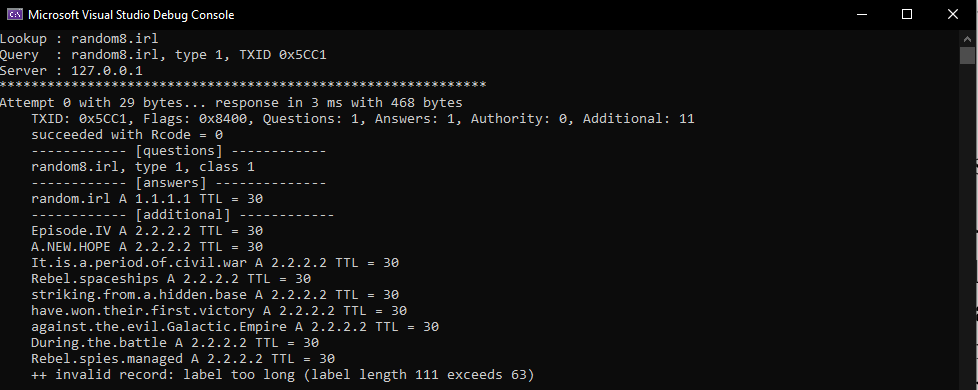
**For example: ++ invalid record: label too long (label length 111 exceeds 63)**

This error is caught using the following check in the code:

A screenshot of a computer

AI-generated content may be incorrect.

This check enforces the **RFC 1035** standard ([RFC 1035 - Section 3.1](https://datatracker.ietf.org/doc/html/rfc1035)), which limits individual DNS labels to **a maximum of 63 bytes**. Since random8.irl generates unpredictable responses, some labels exceed this limit, causing the error to be detected and reported by our parser.



Others could be checked with multiple checks like these for example:  
A screen shot of a computer program

AI-generated content may be incorrect.

Example on how do responses with “lol’s” could look like:  
A screenshot of a computer screen

AI-generated content may be incorrect.



A screenshot of a computer screen

AI-generated content may be incorrect.

