A binary executable and a network capture (.pcapng) file were provided. The challenge involved analyzing a covert communication channel that exfiltrates data through timing intervals between network packets.

The binary file implements a timing-based data exfiltration tool that:

1. Reads the file secrets.txt containing binary data (only '0' and '1' characters)
2. Sends a TCP SYN packet for each character in the file
3. Encodes each character as a timing delay between packets:
   Character '0' → 100ms delay
   Character '1' → 500ms delay

```
while (var_154 < strlen(rax_1))
{
    send_tcp_syn(rax_4, "192.168.184.136", "127.0.0.1", 0x3039, 0x50);

    if (rax_1[var_154] == 0x30)
        usleep(0x186a0);
    else if (rax_1[var_154] == 0x31)
        usleep(0x7a120);

    var_154 += 1;
}
```

4. Targets 127.0.0.1:80 (localhost) from source 192.168.184.136:12345

So, we have to extract the timestamps from out PCAPNG file :

tshark -r capture.pcapng -Y "tcp.srcport==12345" -T fields -e frame.time_epoch > timestamps.txt

The timestamps should look like this :

1741275906.624863253
1741275906.725059919
1741275907.225377289
1741275907.725535233
…

All we have to do is to calculate the inter-packets intervals:

1741275906.624863253
1741275906.725059919 # Diff: 0.100196666s ≈ 100ms
1741275907.225377289 # Diff: 0.500317370s ≈ 500ms
1741275907.725535233 # Diff: 0.500157944s ≈ 500ms

For 100ms -> ‘0’, for 500ms -> ‘1’

Here is the Python code for automatisation :

```
timestamps = [
    1741275906.624863253,
    1741275906.725059919,
    1741275907.225377289,
```

```
    1741275907.725535233,
    1741275907.825811504,
    ...
]
bits = []
for i in range(1, len(timestamps)):
    diff = timestamps[i] - timestamps[i-1]
    if 0.08 < diff < 0.15:  # ~100ms
        bits.append('0')
    elif 0.45 < diff < 0.55:  # ~500ms
        bits.append('1')
    else:
        bits.append('?')  # debugging

binary_str = ''.join(bits)
while len(binary_str) % 8 != 0:
    binary_str += '0'

message = ''
for i in range(0, len(binary_str), 8):
    byte = binary_str[i:i+8]
    message += chr(int(byte, 2))

print(message)
```

THE FLAG : ctf{baf9eecbff31e3d0fcdf9dc820d673594261100b0556d764800bc752d63ecdae}
~Z4que