# CENG519 - Phase 3 Report

Cansu Eskici
2588036

June 15, 2025

## Introduction

My choice of covert channel was *Using options fields in TCP headers (such as timestamps) for data hiding.* In this phase, I focused on detecting the covert channel that I implemented in the previous phase.

The detection mechanism is based on analyzing the TCP header options, specifically the timestamp values. By monitoring the timestamp options in the TCP headers, patterns or anomalies that may indicate the presence of hidden data are identified. The *CovertChannelDetector* class in *tcp-options-processor* implements this detection logic by capturing and inspecting TCP packets for specific timestamp values that correspond to the covert channel's encoding scheme.

## Detection Algorithm

The detection algorithm can be summarized in the following steps:

1. Capture TCP packets from the network interface.
2. Extract the TCP header options, focusing on the timestamp values.
3. Analyze the timestamp values for patterns or anomalies.
4. If a potential covert channel is detected, log the details for further analysis.

The detector passively observes TCP packets and extracts the timestamp values from the TCP options. It maintains a buffer of timestamp values which gets analyzed after receiving a termination signal (a timestamp value of 0). Detection process involves analyzing the entropy of the least significant bits (LSBs) of the timestamp values. For each possible LSB bit, the detector computes the entropy. In a normal scenario, the entropy increases steadily as more bits are considered. However, if a covert channel is present, the entropy will plateau or increase less rapidly at the bit width used for hiding data, since those bits are manipulated to encode the hidden message. The **entropy-threshold parameter** is used to automatically select the optimal LSB bit width by searching for a plateau in the entropy curve, indicating the presence of a covert channel. If a plateau is found and the entropy ratio is sufficiently low, the detector logs the timestamp values and the detected bit number, indicating the presence of a covert channel.

By this approach, the detector automatically infers both the presence of a covert channel and the number of bits used for encoding, without prior knowledge of the sender's configuration. All detection steps are performed passively, without disrupting the normal flow of network traffic.

## Results

Table 1: Detection and correct detection rates for various configurations.

| Bits | Entropy | Message Length | Detector Rate | Correct Rate |
|---|---|---|---|---|
| 3 | 0.10 | 21 | 0.67 | 0.00 |
| 3 | 0.10 | 75 | 1.00 | 0.00 |
| 3 | 0.10 | 141 | 1.00 | 0.00 |

Continued on next page

**Table 1 Covert Channel Detection Results**

| Bits | Entropy | Message Length | Detector Rate | Correct Rate |
|---|---|---|---|---|
| 4 | 0.10 | 21 | 1.00 | 0.00 |
| 4 | 0.10 | 75 | 1.00 | 0.00 |
| 4 | 0.10 | 141 | 1.00 | 0.00 |
| 6 | 0.10 | 21 | 1.00 | 0.00 |
| 6 | 0.10 | 75 | 1.00 | 0.00 |
| 6 | 0.10 | 141 | 1.00 | 0.00 |
| 8 | 0.10 | 21 | 1.00 | 0.00 |
| 8 | 0.10 | 75 | 1.00 | 0.00 |
| 8 | 0.10 | 141 | 1.00 | 0.00 |
| 11 | 0.10 | 21 | 1.00 | 0.00 |
| 11 | 0.10 | 75 | 1.00 | 0.00 |
| 11 | 0.10 | 141 | 1.00 | 0.00 |
| 16 | 0.10 | 21 | 1.00 | 0.00 |
| 16 | 0.10 | 75 | 1.00 | 0.00 |
| 16 | 0.10 | 141 | 1.00 | 0.00 |
| 3 | 0.20 | 21 | 0.67 | 0.00 |
| 3 | 0.20 | 75 | 1.00 | 0.00 |
| 3 | 0.20 | 141 | 1.00 | 0.00 |
| 4 | 0.20 | 21 | 1.00 | 0.00 |
| 4 | 0.20 | 75 | 1.00 | 0.00 |
| 4 | 0.20 | 141 | 1.00 | 0.00 |
| 6 | 0.20 | 21 | 1.00 | 0.00 |
| 6 | 0.20 | 75 | 1.00 | 0.00 |
| 6 | 0.20 | 141 | 1.00 | 0.00 |
| 8 | 0.20 | 21 | 1.00 | 0.00 |
| 8 | 0.20 | 75 | 1.00 | 0.00 |
| 8 | 0.20 | 141 | 1.00 | 0.00 |
| 11 | 0.20 | 21 | 1.00 | 0.00 |
| 11 | 0.20 | 75 | 1.00 | 0.00 |
| 11 | 0.20 | 141 | 1.00 | 0.00 |
| 16 | 0.20 | 21 | 1.00 | 0.00 |
| 16 | 0.20 | 75 | 1.00 | 0.00 |
| 16 | 0.20 | 141 | 1.00 | 0.00 |
| 3 | 0.30 | 21 | 0.67 | 0.00 |
| 3 | 0.30 | 75 | 1.00 | 0.00 |
| 3 | 0.30 | 141 | 1.00 | 0.00 |
| 4 | 0.30 | 21 | 1.00 | 0.00 |
| 4 | 0.30 | 75 | 1.00 | 0.00 |
| 4 | 0.30 | 141 | 1.00 | 0.00 |
| 6 | 0.30 | 21 | 1.00 | 0.33 |
| 6 | 0.30 | 75 | 1.00 | 0.00 |
| 6 | 0.30 | 141 | 1.00 | 0.00 |
| 8 | 0.30 | 21 | 1.00 | 0.00 |
| 8 | 0.30 | 75 | 1.00 | 0.00 |
| 8 | 0.30 | 141 | 1.00 | 0.00 |
| 11 | 0.30 | 21 | 1.00 | 0.00 |
| 11 | 0.30 | 75 | 1.00 | 0.00 |
| 11 | 0.30 | 141 | 1.00 | 0.00 |
| 16 | 0.30 | 21 | 1.00 | 0.00 |
| 16 | 0.30 | 75 | 1.00 | 0.00 |
| 16 | 0.30 | 141 | 1.00 | 0.00 |
| 3 | 0.40 | 21 | 0.67 | 0.00 |

Continued on next page

**Table 1 Covert Channel Detection Results**

| Bits | Entropy | Message Length | Detector Rate | Correct Rate |
|---|---|---|---|---|
| 3 | 0.40 | 75 | 1.00 | 0.00 |
| 3 | 0.40 | 141 | 1.00 | 0.00 |
| 4 | 0.40 | 21 | 0.33 | 0.00 |
| 4 | 0.40 | 75 | 0.67 | 0.00 |
| 4 | 0.40 | 141 | 1.00 | 0.00 |
| 6 | 0.40 | 21 | 1.00 | 0.67 |
| 6 | 0.40 | 75 | 1.00 | 0.33 |
| 6 | 0.40 | 141 | 1.00 | 0.00 |
| 8 | 0.40 | 21 | 1.00 | 0.33 |
| 8 | 0.40 | 75 | 1.00 | 0.00 |
| 8 | 0.40 | 141 | 1.00 | 0.00 |
| 11 | 0.40 | 21 | 1.00 | 0.00 |
| 11 | 0.40 | 75 | 1.00 | 0.00 |
| 11 | 0.40 | 141 | 1.00 | 0.00 |
| 16 | 0.40 | 21 | 1.00 | 0.00 |
| 16 | 0.40 | 75 | 0.33 | 0.00 |
| 16 | 0.40 | 141 | 0.67 | 0.00 |

# Discussion

The results demonstrate that the entropy-based detector is highly effective at identifying the presence of a covert channel, as indicated by the consistently high detector rates (often 1.00) across all tested configurations of bit width, entropy threshold, and message length. This suggests that the method is robust and sensitive to the statistical anomalies introduced by covert data embedding in TCP timestamp fields. However, the correct detection rate—representing the detector's ability to accurately infer the exact number of LSBs used for encoding—is almost always 0.00, with only a few exceptions (notably at 6 bits and higher entropy thresholds). This highlights a limitation of the current approach: while it reliably signals the existence of a covert channel, it struggles to precisely determine the channel's parameters. This is likely due to the subtlety of entropy plateaus and the inherent noise in real-world network traffic. Trying all the possible bit widths may be possible but it is not computationally efficient. The detector's performance suggests that while it is effective at identifying covert channels, it may require further refinement or additional heuristics to improve its accuracy in determining the specific encoding parameters used by the sender.