Phase 3 Report: Covert Channel Detector Implementation and Evaluation

1. Objective The purpose of Phase 3 is to develop a covert channel detector that identifies data exfiltration attempts over IP packet metadata or packet timing. Two detection methods are implemented:

- IP option inspection (specifically looking for timestamp options)
- Timing analysis based on inter-arrival delays of packets

2. Implementation Details

2.1 Detector Components:

- detect_ip_option(packet): Checks for the presence of the IP option with code 68 (timestamp) in incoming packets.
- detect_timing_pattern(packet, timestamp): Evaluates inter-arrival times (IAT) to detect low-noise, regular timing patterns that suggest a covert communication channel.
- A window of past packet timestamps is used to compute IATs.

2.2 Environment Variables:

- ENABLE_COVERT: Used to determine if covert injection is active.
- MEAN_DELAY: Defines the average packet delay in seconds.
- COVERT_MESSAGE: The ASCII message embedded in timestamp option fields.

2.3 Detection Metric Calculation: Based on ground truth (ENABLE_COVERT):

- True Positive: Covert packet detected
- False Negative: Covert packet missed
- True Negative: Normal packet not flagged
- False Positive: Normal packet flagged

Metrics calculated:

- Accuracy, Precision, Recall, F1-Score

2.4 Covert Injection Simulation: When ENABLE_COVERT is set, characters from the COVERT_MESSAGE are embedded in packets using the timestamp option.

2.5 Statistics The following statistics are also collected:

- Round-trip time (RTT)
- Packet processing delays

- Covert channel capacity (bits/sec)

3. Experimental Setup

- **Packet Flow**: Packets sent using ping from the sec container to the insec container.
- **Processor Setup**: The processor listens on NATS topics inpktsec and inpktinsec, injects covert data, and performs detection.
- **Conditions**: Two separate experiments were conducted:
    - **First run**: Covert disabled (ENABLE_COVERT=0)
    - **Second run**: Covert enabled with COVERT_MESSAGE=HELLO

4. Results and Evaluation

4.1 First Experiment (Normal Traffic, Covert Disabled)

- Ping command: ping -c 100 insec
- Ping summary:

100 packets transmitted, 95 received, 5% packet loss
rtt min/avg/max/mdev = 3.126/4.602/10.034/1.114 ms

- Detection Metrics:

True Positives: 0
False Positives: 0
True Negatives: 95
False Negatives: 0
Accuracy: 100%
Precision: 0
Recall: 0
F1-Score: 0

- Analysis: No false detections, as expected in absence of covert traffic. Detector works correctly for benign packets.

4.2 Second Experiment (Covert Enabled: HELLO)

- Ping command: ping -c 100 insec
- Ping summary:

100 packets transmitted, 72 received, 28% packet loss
rtt min/avg/max/mdev = 2.474/4.142/9.042/0.851 ms

- Detection Metrics:

True Positives: 28
False Positives: 0
True Negatives: 0
False Negatives: 154
Accuracy: 15.38%
Precision: 100%
Recall: 15.38%
F1-Score: 26.67%

- Covert Channel Capacity: 7.56 bits/sec
- Analysis: While all detected packets were covert (precision = 1.0), the detector missed many others (low recall). Detection is limited to timestamp options and consistent timing patterns; improvements could include ML models or frequency domain analysis.

4.3 Third Experiment (Mitigation Applied)

- Ping command: ping -c 100 insec
- Ping summary:

100 packets transmitted, 95 received, 5% packet loss
rtt min/avg/max/mdev = 6.322/10.186/18.565/1.976 ms

- Detection Metrics:

True Positives: 0
False Positives: 0
True Negatives: 0
False Negatives: 207
Accuracy: 0.00%
Precision: 0
Recall: 0
F1-Score: 0

- Covert Channel Capacity: 9.60 bits/sec
- Mitigation Statistics:

Packets Processed: 207
Packets Stripped: 5
Packets Delayed: 195
Packets Dropped: 0
Total Delay Added: 0.563 sec

Average Delay per Packet: 2.72 ms

- Analysis: The mitigation system reduced detectability at the cost of increased RTT. The channel still operated but showed slight improvement in throughput, suggesting trade-offs between stealth and bandwidth.

5. Discussion and Improvements

- The detector performed flawlessly on normal traffic, but its recall on covert traffic was low.
- The mitigation was able to obscure covert patterns, rendering the detector ineffective (0 TP) in this run.
- The timestamp option was not present in all covert packets, likely due to delays in injection or message exhaustion.
- Additional detection methods like entropy analysis or supervised learning could improve performance.

6. Conclusion The Phase 3 detector successfully identifies covert traffic via IP options and timing analysis, but current detection heuristics yield low recall in practical scenarios. With mitigation applied, covert channel detectability dropped entirely, demonstrating effective but performance-costly obfuscation.

Appendices

- See results.json and covert_log.json for full experimental data.
- Source code used is appended or available in the accompanying repository.