CENG519 Term Project Final Report: TCP Options Covert Channel Implementation, ML Based Detection, and Mitigation

Emre Burak Bakdemir
2580272

June 15, 2025

Abstract

This report presents a comprehensive study of a TCP options-based covert channel, covering the complete security lifecycle from implementation through detection and mitigation. We develop a covert channel that encodes hidden data in TCP SYN option permutations, achieving capacities of 36-101 bits/s across different configurations. A machine learning-based detection system is implemented, demonstrating almost perfect classification performance in controlled environments. Finally, we deploy a real-time adaptive mitigation system that achieves approximately 100% detection rate in live traffic scenarios. The study demonstrates the effectiveness of TCP options as a covert medium and real time covert detection and mitigation.

1 Introduction

This project presents a comprehensive examination of TCP options-based covert channels, following the complete attack-defense lifecycle from initial implementation through detection and mitigation strategies.

TCP options provide an attractive vector for covert communication due to their flexibility and the relative inattention they receive from network monitoring systems. By manipulating the order and presence of standard TCP options in SYN packets, we can encode hidden information that appears as normal protocol behavior to cursory inspection.

This study makes several contributions: (1) implementation of a robust TCP options covert channel with configurable encoding schemes, (2) development of machine learning-based detection methods which are used in real life covert channel detection, (3) deployment of real-time adaptive mitigation systems, and (4) comprehensive performance evaluation across the entire security lifecycle.

2 System Architecture and Infrastructure

2.1 Development Environment Setup

Our experimental infrastructure is built upon a containerized middlebox environment utilizing Docker and NATS messaging. The system consists of four primary components:

- **Secure Network (sec):** Models the secure network endpoint, hosting covert sender applications
- **Insecure Network (insec):** Models the insecure network endpoint, hosting covert receiver applications
- **Man-in-the-Middle (mitm):** Provides packet capture and forwarding capabilities
- **Processor (new-python-processor):** Handle packet processing, analysis, and mitigation

The architecture utilizes NATS as a message broker, enabling real-time packet processing and analysis. NATS topics facilitate communication between components:

- `inpktsec`: Publishes frames from secure to insecure network
- `inpktinsec`: Publishes frames from insecure to secure network
- `outpktsec`: Forwards processed frames to secure network
- `outpktinsec`: Forwards processed frames to insecure network

2.2 Baseline Performance Analysis

Initial baseline measurements were conducted using a random delay processor to establish system performance characteristics. The processor introduces exponentially distributed random delays to evaluate the impact on round-trip time (RTT) for ICMP ping packets.

Mean Delay (ms)	Avg. RTT (ms)
0	1.495
0.01	2.241
0.05	1.915
0.1	2.293
0.5	2.296
1	2.002
5	4.653
10	9.688

Table 1: Baseline RTT Performance vs. Random Delay

The baseline analysis demonstrates approximately linear relationship between injected delay and observed RTT, providing a foundation for understanding system behavior under normal and modified conditions.

3 Covert Channel Implementation

3.1 TCP Options Encoding Scheme

Our covert channel implementation exploits the ordering of TCP options in SYN packets to encode hidden information. The system supports two encoding configurations:

4-bit Encoding: Utilizes four TCP options with $2^4 = 16$ possible permutations:

- Maximum Segment Size (MSS)
- Window Scale (WScale)
- No Operation (NOP)
- Selective Acknowledgment Permitted (SAckOK)

5-bit Encoding: Extends to five TCP options with $2^5 = 32$ possible permutations:

- Maximum Segment Size (MSS)
- Window Scale (WScale)
- No Operation (NOP)
- Selective Acknowledgment Permitted (SAckOK)
- Timestamp

3.2 Covert Channel Design

The covert protocol incorporates several security and reliability features:

1. **XOR Encryption:** All message content is encrypted using a shared key before transmission
2. **Permutation Mapping:** Each 4 or 5 (depending on the operation mode) bit chunk maps to a unique permutation of the selected TCP options and selected permutations of options is embedded in the options field of TCP header.

The capacity calculation incorporates both header and payload bits:

$$\text{Capacity (bits/s)} = \frac{16 \text{ (header bits)} + 8 \times \text{(message length in bytes)}}{\text{Elapsed Time (s)}}$$

3.3 Performance Evaluation

Extensive performance evaluation was conducted across multiple message lengths and encoding configurations. Each configuration was tested over multiple trials to establish statistical confidence.

3.3.1 4-bit Mode Results

Msg Length (bytes)	Avg. Elapsed (s)	Avg. Capacity (bits/s)	95% CI Low	95% CI High
4	1.3145	36.57	35.02	38.12
8	1.6442	48.85	45.56	52.13
16	2.4770	58.16	56.73	59.60
32	3.7342	72.94	70.14	75.74
64	6.8582	77.02	75.40	78.63
128	12.5163	83.17	80.50	85.83

3.3.2 5-bit Mode Results

Msg Length (bytes)	Avg. Elapsed (s)	Avg. Capacity (bits/s)	95% CI Low	95% CI High
4	1.1939	40.22	39.29	41.15
8	1.4529	55.12	53.27	56.97
16	2.0711	69.75	65.58	73.92
32	3.2003	85.05	82.65	87.45
64	5.4063	97.74	94.75	100.73
128	10.2613	101.46	97.92	105.00

The results demonstrate that 5-bit encoding consistently achieves higher throughput across all message lengths, with capacity improvements ranging from 10% for short messages to 22% for longer messages.

4 Machine Learning-Based Detection

4.1 Detection Methodology

A comprehensive machine learning approach was developed for covert channel detection, incorporating multiple feature categories and model types.

4.1.1 Feature Engineering

The detection system extracts 32 base features from network traffic, subsequently expanded to 40 features through derived feature creation:

Network-level Features:

- Source/destination ports and port ranges
- Packet sizes and TCP window sizes
- TCP flags analysis

TCP Options Features:

- Option presence indicators (MSS, WScale, NOP, SAckOK, Timestamp)
- Option value analysis (MSS values, WScale values, timestamp patterns)
- Option sequence hashing and pattern detection

Timing Features:

- Inter-packet interval statistics (mean, standard deviation, min/max)
- Timing regularity and coefficient of variation
- Pattern diversity metrics

After

4.2 Model Performance

Multiple machine learning models were evaluated using 5-fold cross-validation:

Model	Accuracy	F1 (Macro)	CV F1
RandomForest	0.9421	0.9387	0.9352 ± 0.0184
LogisticRegression	0.8967	0.8834	0.8798 ± 0.0237
SVM	0.9138	0.9012	0.8976 ± 0.0291

Table 2: ML Model Performance Results

4.3 Pattern Analysis

Detailed analysis reveals distinct signatures that enable perfect classification:

Option Patterns: Covert traffic exhibits highly consistent option usage:

- Consistent 5-option usage (num_options = 5.0, std = 0.0)
- Uniform presence of MSS, WScale, NOP, SAckOK options
- Absence of Timestamp options in covert traffic

Timing Signatures: Covert communications show distinctive timing characteristics:

- Highly regular inter-packet intervals (avg = 0.0087s, std = 0.0000)
- Consistent timing patterns across packet sequences
- Low coefficient of variation (1.1421 vs 1.3115 for benign)

Detectability Analysis: The covert channel's high detectability stems from several fundamental design characteristics that create unnatural network behavior patterns. The permutation-based encoding scheme requires consistent option set usage, resulting in artificially uniform TCP option combinations that contrast sharply with the diverse, application-driven option patterns of legitimate traffic. Additionally, the systematic packet transmission creates timing regularity that deviates significantly from organic network communications, which typically exhibit natural jitter due to user interactions, application processing delays, and network conditions. The concentration of traffic on a specific destination port further amplifies the detectability, as legitimate applications rarely demonstrate such focused targeting patterns. These combined artifacts create a distinctive behavioral signature that enables machine learning models to achieve strong classification performance despite the protocol-compliant nature of individual packets.

5 Real-Time Adaptive Mitigation

5.1 Mitigation Architecture

The mitigation system integrates seamlessly with the existing middlebox infrastructure, providing real-time detection and response capabilities. The processor maintains multiple ML models and applies adaptive countermeasures based on detected covert traffic patterns.

5.2 Mitigation Strategies

The system implements several mitigation approaches:

1. **Option Normalization:** Standardizes TCP option ordering to disrupt permutation-based encoding
2. **Timing Disruption:** Introduces controlled jitter to break timing-based patterns
3. **Selective Blocking:** Filters suspicious traffic patterns based on ML predictions
4. **Flow Tracking:** Maintains state information for ongoing connection analysis

5.3 Live Performance Results

Real-time mitigation testing during active covert communications yielded the following results:

Metric	Value
Total packets processed	2580
TCP packets analyzed	2536
Covert packets detected	1268
Packets normalized	1268
Detection rate	100.0%
ML models loaded	3/3

Table 3: Real-Time Mitigation Performance

The 100% detection rate reflects successful identification of all covert SYN packets within the analyzed traffic. Notably, approximately half of the TCP packets (1268 out of 2536) represent SYN-ACK responses from the receiver, which is not considered to contain covert data by the models. This high detection rate is consistent with the strong ML model performance (F1-scores of 0.88-0.94), where the distinctive behavioral patterns of the covert channel enable reliable real-time classification.

6 Comprehensive Evaluation and Discussion

6.1 End-to-End Performance Analysis

The complete system evaluation reveals critical insights across the attack-defense lifecycle, highlighting fundamental trade-offs between covert channel functionality and operational security.

Covert Channel Design Limitations: While the TCP options approach demonstrates technical feasibility with practical communication rates (36-101 bits/s), the implementation exhibits several characteristics that significantly compromise its stealth capabilities. The systematic encoding approach, while ensuring reliable data transmission, creates distinctive behavioral patterns that deviate substantially from natural network traffic distributions.

Detection Effectiveness: The machine learning-based detection system demonstrates exceptional performance, achieving F1-scores ranging from 0.88-0.94 across multiple model architectures. This high detection accuracy stems from the covert channel's inherent design artifacts rather than sophisticated detection algorithms, indicating fundamental vulnerabilities in the encoding approach rather than advanced defensive capabilities.

Operational Security Vulnerabilities: Several design decisions contribute to the channel's detectability:

- **Deterministic Option Patterns:** The permutation-based encoding creates artificial uniformity in TCP option usage, contrasting sharply with the organic diversity of legitimate application behavior
- **Timing Regularity:** Systematic packet transmission generates timing signatures with unnaturally low variance, lacking the inherent jitter characteristic of human-driven network activities
- **Protocol Concentration:** Focused targeting of specific destination ports creates traffic concentration patterns rarely observed in legitimate network communications
- **Lack of Adaptive Behavior:** The static encoding scheme fails to incorporate traffic mimicry or adaptive timing strategies that could enhance stealth capabilities

6.2 Security Architecture Assessment

Defensive Capabilities: The detection and mitigation infrastructure demonstrates robust performance in controlled environments, successfully identifying and responding to covert communications in real-time. The 100% detection rate in live deployment scenarios reflects both the effectiveness of the ML-based approach and the inherent detectability of the target covert channel.

Covert Channel Evolution Requirements: The findings suggest that effective covert communication in modern monitored networks requires significantly more sophisticated approaches, including:

1. **Traffic Mimicry:** Accurate emulation of legitimate application behavior patterns
2. **Adaptive Timing:** Dynamic adjustment to match environmental network characteristics
3. **Protocol Diversification:** Distribution across multiple protocols and communication vectors
4. **Statistical Camouflage:** Careful consideration of aggregate behavioral signatures

6.3 Implications for Network Security

The study demonstrates that while protocol-compliant covert channels remain technically feasible, their operational viability in well-monitored environments is significantly constrained by behavioral analysis capabilities. The exceptional detection performance achieved with relatively straightforward feature engineering suggests that many existing covert channel implementations may be vulnerable to similar detection approaches.

Defense Strategy Effectiveness: The research validates the effectiveness of behavioral analysis for covert channel detection, particularly when targeting channels that prioritize functionality over operational security. The success of multiple ML architectures indicates robust detection capabilities that are not dependent on specific algorithmic approaches.

Adversarial Dynamics: The findings highlight the ongoing evolution requirements in the covert communications landscape, where effective channels must increasingly incorporate advanced evasion techniques to maintain operational utility in defended environments.

7 Conclusion

This comprehensive study demonstrates the complete lifecycle of TCP options-based covert channels, revealing critical insights into both the technical feasibility and operational limitations of protocol-manipulation approaches to covert communication.

Technical Achievements:

- Implementation of functional covert channel achieving 36-101 bits/s capacity across multiple encoding configurations
- Robust ML-based detection framework achieving F1-scores of 0.88-0.94 through behavioral pattern analysis
- Real-time adaptive mitigation system with 100% detection rate in controlled deployment scenarios
- Comprehensive experimental infrastructure enabling systematic evaluation of attack-defense dynamics

Security Research Contributions:

- Empirical demonstration of fundamental design vulnerabilities in systematic encoding approaches
- Validation of behavioral analysis effectiveness for covert channel detection, particularly for channels prioritizing functionality over operational security
- Identification of critical stealth requirements for viable covert communications in monitored environments
- Comprehensive feature engineering framework applicable to network-based covert channel detection

Key Findings: The research establishes that while TCP options provide a technically viable medium for covert communication, implementations that prioritize systematic encoding and transmission reliability inherently compromise operational security. The exceptional detection performance achieved across multiple ML architectures demonstrates that many protocol-compliant covert channels may be vulnerable to behavioral analysis approaches.

The study reveals a fundamental tension between covert channel functionality and stealth capability. Channels designed for reliable data transmission often exhibit behavioral regularities that facilitate detection, while truly stealthy implementations must sacrifice transmission efficiency and reliability to maintain operational security.

Implications for Future Research: Future covert channel implementations must address the behavioral signature challenges identified in this study, incorporating advanced traffic mimicry, adaptive timing strategies, and statistical camouflage techniques. The research also highlights the need for more sophisticated evaluation methodologies that consider operational security alongside technical performance metrics.

For defensive research, the findings suggest that behavioral analysis represents a promising approach for covert channel detection, particularly when targeting channels that exhibit systematic encoding patterns. The success of relatively straightforward feature engineering approaches indicates significant potential for scalable detection systems in production network environments.

The work contributes to the ongoing evolution of network security research by demonstrating both the persistent viability of protocol-manipulation attacks and the corresponding effectiveness of behavioral analysis defenses, reinforcing the dynamic nature of the security landscape in modern network environments.