Network Covert Channels and Countermeasures: A Survey

by Ghanashyam Mahesh Bhat

Submission date: 09-Mar-2023 07:05PM (UTC+0530)

Submission ID: 2032961999

File name: Network_Covert_Channels_and_Countermeasures_A_Survey.pdf (108.14K)

Word count: 3300

Character count: 18799

Network Covert Channels and Countermeasures: A Survey

Ghanashyam Mahesh Bhat
Dept of Computer Science
PES University
Bengaluru, India
ghanashyambhat6@gmail.com

Karan Bhat Sumbly
Dept of Computer Science
PES University
Bengaluru, India
karansumbly7@gmail.com

Kumkum Geervani
Dept of Computer Science
PES University
Bengaluru, India
kumkum1geervani@gmail.com

Prajwal Bhat
Dept of Computer Science
PES University
Bengaluru, India
prajju1205@gmail.com

Dr. Sapna V M
Dept of Computer Science
PES University
Bengaluru, India
sapnavm@pes.edu

***Abstract*—Unidentified covert channels in a network can seriously compromise security, especially in high-security environments where sensitive information needs to be shielded from unauthorized access. The common techniques used to counter covert channel are monitoring, filtering, and encryption but the effectiveness of these defense mechanisms depends on the network's ability to identify and anticipate the covert channels used by the attacker. In our work, we have surveyed various covert channel classes and their detection schemes available across the literature. We have also devised some prevention schemes to obstruct timing network covert channel communication.**

***Index Terms*—covert channel, machine learning, malware, data exfiltration, network, detection, prevention, dataset generation**

I. Introduction

A covert channel is said to be a secret communication channel between two hosts that is utilized to obfuscate data and get beyond security precautions. The fleeting headway in computing power provides new opportunities in expanding the use of covert channel communications for malware exfiltration of data, orchestrating nodes of a botnet, or sending remote commands[2] while avoiding getting detected by firewalls, intrusion detection systems, and anti-viruses [11]. Although covert channels can be used for achieving sound objectives like implementing digital watermarking in VoIP traffic, developing traceback techniques, or avoiding censorship, in reality, it is being used for criminal activities making them a threat to cybersecurity [12]. Covert channels pose a serious risk to the privacy and security of the systems since it goes mostly undetected because of the least generalizability and scalability of the covert channel detection and analysis tools [2]. The issue arises from the fact that there are indefinite ways to conceal information in a covert channel and the tools developed can only address certain types of covert channels. This paper sheds light on the details pertaining to the various types of covert channels that are available across literature and techniques for the detection of the same along with some approaches that can be used to hinder covert communication. The objective of this literature review is to provide a base to anyone interested in developing mechanisms/tools to prevent covert channel communication.

II. Covert Channel Classes

Storage, Timing, and Hyper covert channels [1] are the three main and most popular categories of network covert channels.

A. Storage Channels

Storage channels operate under the tenet of directly obscuring information in the traffic stream[2] by altering the header or payload fields of the protocols. This implies amending the packet structure, altering the padding bits, or re-arranging optional fields[6]. This can be achieved by altering the packet header fields like Type of Service, and Time to Live or by modifying packet length. These covert channels are popular due to their high bandwidths[9].

There are two distinct types of Network Storage Covert Channels: Payload Covert Channels and Non-Payload Covert Channels. Payload Covert Channels are used to hide the covert message in the payload area of the protocol whereas in Non-Payload Covert Channels we modify the header fields[9]. Storage covert channels can be detected and analyzed using approaches based on Condensed Statistical Indicators[6], and tools like Bccstego[6].

Storage channels do not demand specialized hardware or software for their purpose. They are a practical and affordable choice for attackers. Yet another reason for attracting the attackers is their ability to transfer high amounts of data rapidly in a short span of time.

B. Timing Channels

Timing covert channels are another type of covert channel that takes advantage of differences in timing events to transmit information between two parties. The covert message is programmed into an entity's timing behaviour. A network timing covert channel transmits secret messages by altering the timing of network packets. There are two types of timing channels: active and passive timing channels. The sender purposefully delays the delivery of specific packets on an active timing channel to communicate a hidden message. For example, the sender may introduce a 10-millisecond delay to every fifth packet, which the receiver would interpret as the secret message.

The sender does not purposefully delay packets in a passive channel but instead leverages the network's inherent timing fluctuations to communicate a hidden message. For example, to convey a secret message, the sender may only transmit packets at particular time intervals, and the recipient would interpret these time intervals as the secret message.

In a passive timing channel, the sender will not delay packets actively, but instead, he will use the network's inherent timing fluctuations to deliver a covert message. For instance, to deliver any secret message, the sender may target specific time intervals in the network and may only transmit packets during the same.

According to the authors in [1], due to overlaps in the time frames of both covert and overt(open) traffic, it will be complex to differentiate between the two if the maximum limit for packet delays to hide covert messages is equal to or falls short of one-fourth of the average of the inter-arrival times of the overt traffic. A few instances of covert channels that make advantage of network timing include altering timings for requests and responses for Domain Name System(DNS), Internet Control Message Protocol (ICMP) echo messages, and acknowledgments in Transmission Control Protocol(TCP).

Timing channels work efficiently as, instead of adding new data, they manipulate the time of existing traffic making the detection difficult even after using intrusion detection tools. When it comes to applications for interactive communication (HTTP, SSH), small time variations make it easier to insert more data by simply altering the time characteristics of packets[5]. They have low bandwidth[9] and are hence less popular.

C. Hyper Channels

Timing and storage channels are combined in the concept of a hyper-covert channel. Timing channels are relatively challenging to discover, however, storage channels have an advantage in delivering soaring bandwidth[3]. Hyper channels integrate both these advantages, hence detection of hyper channels is very laborious. Using high-frequency sound waves to transport data, including data in seemingly innocent data like photos or video files are a few examples of hyper channels. Detecting these channels can be exceedingly difficult without sophisticated analysis tools, and may require specialized expertise and experience.

Although we have classified covert channels into three main categories, there are indeed numerous ways to create a covert channel. The first release of the CCgen tool for the injection of the covert channel itself provides multiple ways of injecting covert channels into real-world traffic[12]. The diversity of covert channels makes it difficult to generalize the solution to detect covert communication effectively while making sure that it doesn't add much overhead to the network[13].

III. Dataset Generation

As there aren't many publicly available datasets, obtaining public network traffic for covert channel identification might be challenging. Current datasets have various issues including uneven distribution and out-of-date content and creating such network traffic datasets can be time intensive[1]. During our survey, however, we came across some datasets available publicly like NSL-KDD[1], and the Defense Advanced Research Projects Agency(DARPA) Dataset which consists of a variety of covert communication-related network traffic scenarios. We can also produce synthetic data which imitates the traits of covert channels using traffic generators, synthetic tools like NS-3 or OPNET, and reverse engineering protocols. One can also create a simulation of covert channels in virtual network environments to produce datasets using software and emulators such as GNS3 or EVE-NG. Network traffic captured from actual circumstances can also be used to create a dataset for covert channels. We can recognize several kinds of covert channels and note their properties by examining network traffic. Protocol tunnelling entails enclosing one protocol inside another in order to create covert channels. Instead of using software-based simulations, we can employ Hardware-in-the-Loop (HIL), which makes use of routers and switches in order to increase accuracy.

We can make use of Python, Wireshark, Scapy[1], Covertutils and Cryptcat to aid in dataset generation. In [5], we have observed the usage of dataset containing TCP traffic from LBNL/ICSI Enterprise Tracing. Channels containing 4 and 8 bits symbols are used for dataset creation[5]. Another instance is seen from the traffic obtained on an OC192 link given by CAIDA[6]. Datasets may contain images, text documents, encoded files,compressed data[8], etc. pcapStego is a great tool to build large real network traffic datasets contained inside .pcap files targeting Flow labels, TTL value and Traffic class. Another real network dataset is avaialable with the help of MAWI group. The most common approach to solving problems due to covert channels is using machine learning techniques, which require a large volume of data for training the model[10].

Collecting the data from real traffic for the purpose of research is not ethical since the data in a real-world network will be private and may contain sensitive information[10]. Datasets generated by tools have proved to be far more efficient than any other toy datasets[10]. Hence it is necessary to generate our own dataset with the help of the existing tool which mimics real-world traffic.

IV. Analysis of Covert Channels

There are various analysis techniques available for covert channels. Analysis methods must be fast and lightweight.

In DAT detectors[5], Multimodality and the Sum of Autocorrelation coefficients are used.

- Multimodality - 1) Uses Kernel density distributions for the estimations using Pareto charts. We can gauge multimodality using the statistical modes obtained from the data. Statistical modes are the total number of notable peaks obtained from the probability density function of arbitrary variable.
  2) Symbols using Pareto Charts - This aims to obtain the count of all those symbols which recur most with the help of Pareto analysis.
- Sum of Auto-correlation coefficients - Covert channels have a greater probability of being detected by this method.

Graphical Analysis of covert channel pattern is most commonly seen. Inter-field analysis makes use of a set of patterns gained from ML classifiers at the time of offline analysis. It considers combinations of OD-flow vector values in various TCP/IP header fields[5].

A set of customized rules and thresholds are defined for every specific field according to the expected distribution characteristics, field type, and value ranges done in intra-field analysis. It also helps in checking TCP/IP header fields separately by examining the corresponding OD-flow vector values[3]. Sensitivity analysis is one such method helpful in detecting storage channels. Tools like Wireshark are used for the analysis of packets flowing over a network[7].

bccStego is a high-throughput tool for measuring general statistics of the packets[11] where the state is neglected in order to achieve lesser memory usage and minimal overhead. This tool can be integrated with other frameworks to design the detection scheme with less overhead to the network. bccStego makes use of BPF and it can be extended to measure different aspects of the packets as per the requirement. Once the data is collected, the presence of the covert channel can be detected by analyzing the variation in the bin size[11].

V. Detection Schemes

The switching techniques assist in developing a covert channel that is hard to detect[3]. Internal control protocol technique that provides a reliable and trusted communication channel for a covert message[3] increases the use of these channels.

Challenges faced during the covert channel detections include: a) Noisy Channels b) Covert channels in address fields and bouncing covert channels c) Covert channels modeled according to field distribution properties d) Covert channels with encrypted messages. Storage channels provide high bandwidth and timing channels are hard to detect[3]. Hybrid covert channels pose a real challenge in detection since they are a combination of storage and time channels.

We can use DAT detectors, various Machine Learning approaches,etc.

Covert channel Detection techniques can be classified as:

- Value to Symbol correspondence - usually detected by the intra-field analysis. The context abstraction is compared with references of expected or non-suspicious distributions, which are ultimately shaped by a set of rules and thresholds[5].
- Value range as Symbols - similar to Value to symbol correspondence but the number of multimodality distribution peaks is more than the number of multimodality symbols[5].
- Container fields - The intra-field analysis block is mainly addressed to checking container fields in the IP and TCP options fields. They can be detected using packet compliance checks [5].
- Timing Channel - can be considered as value ranges as symbols for the packet IAT field[5].
- Derivative Approaches - detected using the sum of auto-correlation coefficients and mean differences[5].

Detection methodologies in Passive Warden Scenario can be grouped into[5]:

- Traffic irregularities – Packet Compliance Checking Phase
- Statistical Analysis – Multimodality, Autocorrelation, Descriptive Statistics
- Machine Learning – Inter-field Analysis(Offline Analysis)

Detection of Storage Covert Channels includes:

- Detection of Channels Targeting the Flow Label
- Sensitivity Analysis
- Channels Targeting other Ipv6 fields[5]

Sensitivity analysis includes three attack scenarios[6]:
a) Exfiltration attempt modeled via transmission of file requiring to target 8500 Ipv6 packets.
b) Different channels alter in time.
c) APT targeting datacenter or subnetwork.

The test environment for the covert channel detection and prevention system consists of a set of multiple virtual machines (VMs), one of which serves as the server, while the others function as clients. The clients exfiltrate data through a timing covert channel that the server receives and decodes[14]. The covert channel operates using predetermined delays for setting and unsetting bits. The server analyzes the packet reception time to detect covert messages. It is worth noting that even slight variations in the delay and pattern of the covert channel can go undetected by normal intrusion detection systems (IDS). Therefore, a machine learning (ML) model needs to be trained to detect various patterns to identify the timing channel, even if the channel is implemented using different methods. This approach ensures that the covert channel detection and prevention system is capable of detecting any attempts to transmit data through the timing covert channel, regardless of the techniques used to conceal the communication.

With the help of a dataset, one can devise a machine learning model to detect covert communication and thus assist in taking actions to prevent the same. SVM has shown a significant result in detecting covert channels over other algorithms, though the statistical variations between the covert channels and overt channels must be clear[1].

With the fast-growing technology and a wide scope for covert channel generation, the machine learning model has to be retrained periodically to detect advanced covert channels[1].

VI. Prevention Schemes

Based on the literature we have surveyed on covert channels, we were able to devise some techniques that could be effective against network covert channel communication. However, we limit our focus to the preventive schemes for Timing-Based Covert channels.

In [8], the authors shed light on various timing techniques that can be used for network covert communication.

- Packet presence Technique: For this method to work, the sender and receiver must be in sync and agree on a set time period for sampling. During a certain interval, a packet's presence or absence translates to a binary 0 or 1 correspondingly.
- Fixed intervals: In order to represent 0s and 1s, this approach establishes defined inter-departure times for packets.
- Jitterbug: The Jitterbug modifies a pre-existing network transmission. In order to create packet inter-departure times divisible by X or $X/2$, depending on the covert symbol to broadcast, it sets a base sampling time interval of let's say X and adds a little delay to them. Divisibility by X or $X/2$ may be used to represent binary bits 0 or 1 respectively or vice-versa.
- Huffman encoding: This approach is based on the Huffman compression algorithm to directly encode each covert symbol into a group of packets with various packet inter-departure times dependent on the frequency of the encoded symbol.
- One threshold: This approach defines a threshold X for packet inter-arrival periods, especially for Android systems with video services. Delays in packet arrivals are recorded as 1s or 0s, depending on if the delay value is less than or greater than X.
- Packet bursts: This approach generates packet bursts that are separated apart by a waiting period say, X. The hidden symbol or piece of code to transmit is directly defined by the number of packets in a burst.
- Differential/derivative: Every time a '1' is to be transmitted, the basic packet inter-departure time value (idtv) is updated by adding or subtracting t_{diff} from the preceding idtv. If '0' is to be transmitted, the last idtv remains unchanged.

All the above-mentioned timing channel techniques can be potentially disrupted by adding a random buffer/delay to the packets by the network router firewall. Apart from those mentioned above, one other timing technique as mentioned in [8] is:

- Timestamp manipulation: Using this method, the least significant bit (LSB) of the TCP timestamp is used to alter packet inter-departure times. The authors propose that the inter-departure time between packets be at least tb = 10ms. The packet is transmitted if the LSB matches the covert symbol; else, the condition is rechecked after a delay of $twait$. This technique can be countered if we configure the router in a way that increments/ decrements the last 3-4 bits of the timestamp field of the packet headers. However, this random change in timestamp must be synchronized with other packets of the same logical connection.

VII. Conclusion and Future Work

At the end of our survey, we conclude that the presence of covert channels in a network is the reason for multiple privacy and security concerns. There is a need for effective detection and prevention mechanisms to counter the attacks using covert channels. We have surveyed various covert channel detection schemes present in the literature and suggested some preventive measures with respect to timing channels. Future work consists of devising preventive measures for Storage and Hybrid covert channel types also. It also includes the development of tools based on the suggested detection and prevention schemes that can be placed alongside Intrusion Detection and Prevention Systems in the network.

References

[1] Muawia A. Elsadig, Ahmed Gafar, "Covert Channel Detection: Machine Learning Approaches," vol 10, pp. 38391-38405, 2022.

[2] Marco Zuppelli, Luca Caviglione, Wojciech Mazurczyk, Andreas Schaffhauser, Matteo Repetto, "Code Augmentation for Detecting Covert Channels Targeting the IPv6 FLow Label," 2021, pp. 450-456.

[3] Muawia A. Elsadig, Yahia A. Fadlalla, "Packet Length Covert Channel: A Detection Scheme," 2021 pp. 1-7.

[4] Arti Dua, Vinita Jindal, Punam Bedi, "Covert Communication using Address Resolution Protocol Broadcast Request Messages," 2021, pp. 1-6.

[5] Félix Iglesias, Robert Annessi, T. Zseby, "DAT detectors: uncovering TCP/IP covert channels by descriptive analytics," 2016, pp. 3011-3029.

[6] Marco Zuppelli, Matteo Repetto, Andreas Schaffhause, Wojciech Mazurczyk, "Code Layering for the Detection of Network Covert Channels in Agentless Systems", vol. 19, no. 3, pp. 2282- 2294, 2022.

[7] Punam Bedi, Arti Dua, "ARPNetSteg: Network Steganography Using Address Resolution Protocol," vol. 66, no. 4, pp. 671-677, 2020.

[8] Félix Iglesias, Valentin Bernhardt, Robert Annessi, Tanjab Zseby, "Decision Tree Rule Induction for Detecting Covert Timing Channels in TCP/IP Traffic," *1st International Cross-Domain Conference for Machine Learning and Knowledge Extraction(CD-MAKE)*, pp. 105-122, Aug 2017.

[9] Muawia A. Elsadig, Yahia A. Fadlalla, "Network Protocol Covert Channels: Countermeasures Techniques," 2017.

[10] Marco Zuppelli, Luca Caviglione, "pcapStego: A Tool for Generating Traffic Traces for Experimenting with Network

Covert Channels,” 2021.

[11] Matteo Repetto, Luca Caviglione, Marco Zuppelli, ”bccstego: A Framework for Investigating Network Covert Channels,” 2021.

[12] Félix Iglesias, Fares Meghdouri, Robert Annessi, Tanja Zseby, ”CCgen: Injecting Covert Channels into Network Traffic,” 2022, pp. 1-11.

[13] Marco Zuppelli, Luca Caviglione, Matteo Repetto, ”Detecting Covert Channels Through Code Augmentation,” 2021.

[14] Dewank Pant, Manon Wason, Jibraan Singh Chahal, ”Cross VM Covert Channel Implementation”, 2018.

Network Covert Channels and Countermeasures: A Survey

ORIGINALITY REPORT

14%	8%	13%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	onlinelibrary.wiley.com Internet Source	3%
2	Muawia A. Elsadig, Yahia A. Fadlalla. "Packet Length Covert Channel: A Detection Scheme", 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), 2018 Publication	2%
3	dokumen.pub Internet Source	2%
4	Muawia A. Elsadig, Ahmed Gafar. "Covert Channel Detection: Machine Learning Approaches", IEEE Access, 2022 Publication	1%
5	Shravya Bhat, Shilpa S Nair, Shravya Kadur, Srikanth H R. "A Personalised Approach to Adaptive Tutoring using Machine Learning and Natural Language Processing", 2019 IEEE Bombay Section Signature Conference (IBSSC), 2019 Publication	1%

6	"Machine Learning and Knowledge Extraction", Springer Science and Business Media LLC, 2017 Publication	1%
7	www.annessi.net Internet Source	1%
8	Muawia A. Elsadig, Yahia A. Fadlalla. "Network Protocol Covert Channels: Countermeasures Techniques", 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), 2017 Publication	1%
9	iris.unige.it Internet Source	1%
10	Charvi Bannur, Chaitra Bhat, Gagan Goutham, H. R. Mamatha. "General Transit Feed Specification Assisted Effective Traffic Congestion Prediction Using Decision Trees and Recurrent Neural Networks", 2022 IEEE 1st International Conference on Data, Decision and Systems (ICDDS), 2022 Publication	1%
11	Iglesias, Felix, Robert Annessi, and Tanja Zseby. "DAT detectors: uncovering TCP/IP covert channels by descriptive analytics : DAT detectors: uncovering TCP/IP covert channels by descriptive analytics", Security and Communication Networks, 2016. Publication	<1%

12	Marco Zuppelli, Matteo Repetto, Andreas Schaffhauser, Wojciech Mazurczyk, Luca Caviglione. "Code Layering for the Detection of Network Covert Channels in Agentless Systems", IEEE Transactions on Network and Service Management, 2022 Publication	<1 %
13	de.wikipedia.org Internet Source	<1 %
14	repository.up.ac.za Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		