Conference Paper Title*

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*—This document talks about various covert channel classes, their detection and prevention schemes.**

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

I. Introduction

A covert channel is said to be a secret communication channel between two hosts which is utilised 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. Based on the literature survey, we propose the design and working of a tool that can be developed to detect covert channels.

II. Covert Channel Classes

Storage,Timing and Hyper covert channels [1] are the 3 main and most popular categories of 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 results in amending the packet structure, altering the padding bits, or rearranging optional fields[6].This can be achieved by altering the packet header fields like Type of Service, Time to Live or by modifying packet-length. Storage Covert Channels utilize a mechanism wherein a process writes into a shared storage location directly or indirectly and another process can directly or indirectly read from this location. The processes that utilize this shared storage location can either belong to the same computer or different computers that are communicating over the network. These covert channels are popular due to their high bandwidths[9].

There are two distinct types of 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 disc operations for read and write, system calls, approaches based on Condensed Statistical Indicators[6], and tools like Pcapstego[10].

Storage channels do not demand for specialized hardware or software for their purpose. They are a practical and affordable choice for the attackers. Yet another reason for attracting the attackers is their ability to transfer high amounts of data rapidly in a short span of time. This can be overcome by limiting the access to write to specific shared locations and examining the patterns of read and write.

B. Timing Channels

Timing covert channels are another type of covert channels that take advantage of differences in timing events to transmit information between two parties. There are two types of timing channels: active and passive timing channels[1]. Active timing

Covert channels actively manipulate system activity time to encrypt data whereas Passive timing channels entail using system activity time to decode data. Detection of Active timing channels is easier than that of Passive channels since the latter depends on the receiver's capacity for timing pattern measurement.

In timing covert channels, the covert message is modulated into the timing behavior of an entity[1]. Timing covert channels use a mechanism where the sender utilizes the system's rescore aspect in order to signal a secret message and the receiver observes studies it and decodes the secret message. Timing channels work efficiently as they hide sensitive information using network traffic's timing properties especially packet inter-arrival times[6].

C. Hyper Channels

The idea of a hyper-covert channel combines the ideas of timing and storage channels. 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.

Although we have classified covert channels into three main categories, there are indeed numerous ways to create a covert channel. The first release of 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

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].
It is difficult to get public network traffic for testing covert channel detection since there are not many publicly accessible datasets.[1]. Hence we make use of tools that help in generating datasets. pcapStego is a tool used for creating network covert channels within .pcap files[10]. It aids in the creation of large datasets containing real network traces[10]. targeting Flow Labels, Traffic class, and Hop Limit. The tool allows the creation of channels We can see that data which consists of real TCP traffic taken from the LBNL/ICSI Enterprise Tracing Project[5] is used. A dataset with 4-bit and 8-bit symbol channels is created for testing. An example includes traffic collected on an OC192 link in different conditions made available by the Center for Applied Internet Data Analysis (CAIDA)[6]. The real network data downloaded from MAWI working group traffic is another use case seen. The dataset generated by the tool is better than other toy datasets[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]. Thus making it 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 desnity 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 patterns 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 autocorrelation 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.

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.

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

- Packet presence Technique: This technique requires synchronicity between the sender and the receiver, who agree on a fixed interval as sampling time. The absence or presence of a packet during an interval represents the binary symbol 0 or 1 respectively.
- Fixed intervals: This technique establishes fixed packets' inter-departure times (idts) to represent 0s and 1s.
- Jitterbug: The Jitterbug manipulates an existing transmission. It establishes a ground sampling time interval, let's say X, and adds a little delay to idts to make them divisible by X or $X/2$ according to the covert symbol to send.
- Huffman encoding: This technique codes every covert symbol directly into a set of packets with different idts according to the frequency of the symbol and based on a Huffman codification.
- One threshold: Devised for Android platforms and with video services as carriers, this technique establishes a threshold X for packet inter-arrival times (iats). Delays above and below X will be considered 1s and 0s respectively.
- Packet bursts: By this technique, packet bursts are sent separated by a waiting time interval t. The number of packets in a burst directly represents the covert symbol or piece of code to send.
- Differential/derivative: Given a basic idt, whenever a 1 is to be sent the previous idt is modified by adding or subtracting t_{inc} in case of a 0, the last idt is kept the same, i.e., with no modification.

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

- Timestamp manipulation: By this technique, packet idts are manipulated based on the least significant bit (LSB) of the TCP timestamp. A minimum idt between packets must be respected (authors propose, at least, tb = 10ms). If the LSB coincides with the covert symbol, the packet is sent; if not, the condition is checked again after a time t_wait. 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

We have seen that the covert channel is the reason for multiple privacy and security concerns and there is a need for effective detection and prevention mechanism to counter the attacks using covert channels.

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.