International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.2 (2025) pp. 1910-1920
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Analysis and Mitigation of Covert Timing Communication Channels Using Active Warden Mechanism for Enhancing Network Security

Vrushali Uday Uttarwar [1]*, Dhananjay M. Dakhane[2]

[1] Ramrao Adik Institute of Technology, D.Y. Patil Deemed to be University, Nerul, Navi Mumbai 400706, Maharashtra- India
* **Corresponding Author Email:** uttarwarvrushali@gmail.com - **ORCID:** 0000-0003-3365-4978

[2] Ramrao Adik Institute of Technology, D.Y. Patil Deemed to be University, Nerul, Navi Mumbai 400706, Maharashtra- India
Email: dhananjay.dakhane@rait.ac.in – **ORCID:** 0000-0003-0424-4371

Article Info:

DOI: 10.22399/ijcesen.1484
Received : 12 January 2025
Accepted : 19 March 2025

Keywords :

Covert Communication Channels,
Active Warden Mechanism,
Network Security,
Packet Delay Normalization,
Encoding Schemes.

Abstract:

Covert communication channels (CCC) are very powerful challenges to the network security since they allow for unauthorized data transfer through different ways of mediation, usually avoiding traditional protections. In this study, we study and mitigate such channels through an innovative adaptive warden mechanism. Network traffic is dynamically normalized by the adaptive warden to disrupt covert signals while minimizing its effects on normal traffic data transmission. The mechanisms are evaluated to understand the encoding schemes, including Dual Bit, ON-OFF, and JitterBug, in the context of real-time scenarios. The adaptive warden mitigates CCCs through negligible overhead in transmission times; this is experimentally demonstrated with overhead consistently 1% or less. Moreover, as a contribution, the proposed framework also proposes a scalable approach to mitigating covert communication by means of protocol normalization, delay randomization, and traffic analysis. The study also highlights essential knowledge gaps when addressing CCCs in the IPv6 protocol space, highlighting the demand for improved countermeasures in contemporary network designs. The work presents a general framework for detecting and mitigating CCCs and provides a number of significant advancements in network security. It serves as a basis for future work in adaptive mechanisms to defend against emerging threats to covert communication in heterogeneous and complicated network environments.

1. Introduction

Covert channels as communications paths not intended to transmit information are defined by Lampson [1]. And for secret channels within networks, our main effort focuses on the existence of these hidden channels in local systems. Covert channels in general aim at transmitting information with undetectable intrusion into security policy violation [2]. These hidden routes have been studied by researchers for a long time. Covert channel creation within network packet data and timing has been studied in many papers (e.g., [3-5]). Covert channels are addressed using various methods, such as the pump [6]. Covert timing channels utilizing ACKs are made more limited by the pump limiting the number of acknowledgment messages sent from a higher security level to a lower security level. Additionally, other techniques, such as covert flow trees [7], shared resource matrix (SRM) methods [8], expanded SRM [9], and steganalysis on covert channels in VoIP traffic [10], are also notable. Covert channel disruption utilizing the active warden approach is well established. Instead, inactive wardens monitor and report, i.e., events like detecting intrusions. Network traffic can be modified by active wardens, such as traffic normalizers [11], in order to prevent data transfer of steganographic data [12].

Network covert channels are becoming increasingly significant because of their ability to remain unobserved and the diversity of applications that they have. For example, botnets are secretly controlled using covert channels [13]. Covert channel techniques are also widely utilized by journalists for the free expression of ideas [14]. For

example, in 1997 [15], LOKI2 was introduced that was a covert channel relying on the user to send commands modifying a network protocol. During the last decade, a number of new covert channel techniques have been developed, but many of these have not yet been addressed by security measures. we discuss how these new methods make covert channels exchange their communication protocols transparently and automatically and how they can cooperate in overlay networks via internal control protocols [16]. This paper examines two novel covert channel techniques: protocol switching covert storage channels (also known as protocol hopping covert channels, PHCC) and protocol channels (PCs). Initially PHCC were outlined in [17]. By transmitting confidential information between different network protocols, these channels allow reducing attention to unusual protocol behavior. The "User-Agent" field of HTTP and the "RETR" request of POP3 can be used to send confidential data by a basic PHCC. PCs were originally released and are based on a predefined set of network protocols for transmitting private information [18]. Protocols are therefore associated with hidden values: a POP3 packet may represent the value '0,' but an HTTP packet may indicate the value '1,' Therefore, in order to deliver 110 via this PC, the sender must send two HTTP packets and a POP3 packet. A PC's bandwidth is usually limited to a few hundred bits per second, which is sufficient for an attacker to send a tweet, a selected document, or a password. The first PC (but not PHCC) detection method was developed [19-27]. The bandwidth of PC and PHCC hasn't been reduced or eliminated, however.

1.1. Review of Literature

Chourib, Wendzel, and Mazurczyk [28] proposed an adaptive warden strategy to counter network covert storage channels, addressing the limitations of static and dynamic wardens. Unlike static wardens with fixed normalization rules or dynamic wardens with periodic updates, the adaptive warden selects rules based on traffic characteristics, effectively disrupting covert communication. By increasing the packet requirements for covert data transfer, the strategy provokes covert peers to expose themselves. The study demonstrates that adaptive wardens outperform dynamic wardens in both efficiency and effectiveness, offering a more robust approach to mitigating covert channels.

Due to the increasing reliance on digital communication, there is a never-ending struggle of promoting policy enforcement in controlling communication and giving people privacy and anonymity. Secret communication that is not detected by security mechanisms is done through covert channels operating below the notice of security mechanisms. There are further security challenges in these channels, which are typically buried inside communication protocols. As with the security features of IPv6, adoption of IPv6 is critical, as IPv4 will soon be exhausted; however, the implementation of IPv6 is still underexplored. Covert channels in IPv6 and ICMPv6 are investigated, the protocol tool that measures their performance. Moreover, this study extends active warden concepts to boost covert channel detection and mitigation with network awareness. Significant research efforts have been spent on integrating adaptive covert communication systems into network environments for security and traffic normalization purposes. Recent studies argue for the employment of network interfaces, like DCL and ICL, to execute covert communications and in order to guarantee reliable exchange of traffic. Due to their ability to adaptively change the normalization rules on Confidential Channels, adaptive wardens have come to prominence as concepts. The advantages of considering dynamic rule sets in covert communication systems that disrupt continual flows to reduce the chance of detection by surveillance [29]. Several adaptive warden models, including those based on FIFO (First in First Out) and NRU (Not Recently Used), have been evaluated as randomizing traffic normalization and enhancing the warden's resilience against detection in covert channels [30].

Covert communication about security and undetectable information exchange has motivated extensive research of normalization methods used by firewalls and intrusion detection systems (IDS). The area of focus has been the dynamic warden systems with random rule sets being used to thwart hidden patterns of covert communications [31-34]. Traditional warden methods use static rules that apply a small set of normalization rules, whereas dynamic warden relies on a dynamic set of rules. Using adaptive methods for covert communication, e.g., random rule activation, researchers have demonstrated that traffic is much more difficult to conceal and avoid detection systems [33]. Additionally, the advantage of using tools like Netfilter for normalizing network traffic completes the package of these systems, where studies show one example that adaptive wardens are robust in maintaining covert communications in highly monitored network environments [35]. Central to the evaluation of the performance of adaptive covert communication systems, particularly systems with adaptive wardens, has been their evaluation in real-world scenarios. Often tools like NELphase were used in studies to simulate covert

traffic, examine the communication between the CS and CR, and stress accurate feedback channels (such as ICL) and packet transmission reliability[39]. Finally, counters for packet delivery, memory usage, and CPU load are also included for further performance assessment and tuning of these systems[36]. Generally, the covert communication research is discussed in terms of protocols, such as ICMP, IPv4 SCTP, TCP, UDP, and HTTP, and methods employed for embedding secret contents inside network packets. Adaptive covert communication systems enabled via proof-of-concept implementations with tools such as Libpcap and Scapy have been shown to be practical and flexible across protocols to serve as a basis for continued advancements in information exchange technologies security [31].

1.2. Research Gaps

Despite enormous developments in covert communication systems and countermeasures for such systems, there remain important research gaps in the design of such warden mechanisms for maximizing their effectiveness and efficiency, specifically for covert timing channels. Although Chourib, Wendzel, and Mazurczyk (2021) have made important progress in addressing covert communication over network covert storage channels, the direct application of their adaptive warden strategies to covert timing channels is largely unexplored. Previous work to date has mostly investigated covert storage channels and covert channels in network protocols like IPv6 and ICMPv6 and has not addressed strategies for covert timing channels stemming from the manipulation of packet timing for the transfer of data. Additionally, prior studies have investigated the use of adaptive wardens in dynamic rule-based systems for covert communication control, but no work has been done on the adaptation mechanisms for covert timing channels. The lack of attention paid to covert timing channels is a major limitation, since timing-based covert channels can often bypass the detection mechanisms that are based on static or dynamic traffic normalization. It was shown that adaptive warden systems can provide meaningful performance in real-world scenarios, the effects of active wardens on the latency and throughput of network traffic over covert timing channels remain less understood [30]. There is also yet another area that needs more investigation: the ability to use advanced techniques (e.g., machine learning or AI) to enable automatic detection and counter the use of covert timing channels built into active warden systems. Existing adaptive systems are effective at randomizing traffic normalization, but adaptive warden systems leveraging more general machine learning models may be able to better identify covert timing patterns as they occur in real time, improving the adaptive warden's response to new covert communication strategies.

2. Material and Methods

Our experimental Set Up consist of Docker containers to create the PC instances as presented in [27]. Docker is a virtualization technology and software development tool that facilitates the creation, deployment, and management of programs through the use of containers. A "container" is a tiny, standalone, executable software package that contains all of the libraries, configuration files, dependencies, and other elements needed to run the program. An application and its dependencies can be packaged by Docker and executed in a virtual container on any Linux, Windows, or macOS computer. In other words, because the container provides the environment for the lifetime of the application's software development life cycle, programs work the same regardless of where they are or what computer they are running on. Because of the isolation that containers offer, multiple containers can operate concurrently on a single host. This makes it possible for the application to run in a number of places, including public or private clouds and on-premises. Because Docker containers do not require an additional load from a hypervisor, they are lightweight. Multiple containers can run concurrently on a single server or virtual machine. Since the main objective of the CS and CR is to identify which, active rules are being used by the active warden, we assumed in our evaluation that the NEL phase could be carried out concurrently with the secret data exchange between parties. As soon as the CR received the information on the first formed NCC over the ICL, we started the measurement. We also noted how long it took to figure out nonblocked covert channel strategies and how long it took to send a certain quantity of packets containing concealed data (400 packets, for example) during the communication phase. Once the CR receives the indicated number of packets the measurement is concluded. We evaluated the adaptive warden in the experimental evaluation with varying the twt, ic, and ws parameters. For each parameter, the experiment was run three times and the results are reported as average results were below.

2.1 Experimental Results

This proposed, Covert Channels Evaluation Framework can be used to assess the capacity,

stealth, and robustness of covert channels. It can also be used to test current firewalls or intrusion detection software since it creates covert channels, which can be used to provide the data required to evaluate defences against network covert channels. The sender sends the covert message to the receiver using the developed framework. Jitterbug Covert Channel is as shown in Figure 1. In this, Sender started sending the covert message to the receiver using Jitterbug Covert Channel, the Receiver is signalling the sender to send message and it also shows the receiver has decoded the covert message sent by the sender using the decoding method of SCC. In the figure 1 shows the covert communication between Alice and Bob before enabling the active warden and figure 2. shows the covert communication between Alice and Bob after enabling the active warden. It demonstrates that the active warden running on the gateway is enabled and normalizes the covert communication between Alice and Bob. The receiver actually does not receive the actual covert message, instead it decodes a corrupted message and hence the active warden has eliminated the covert communication.

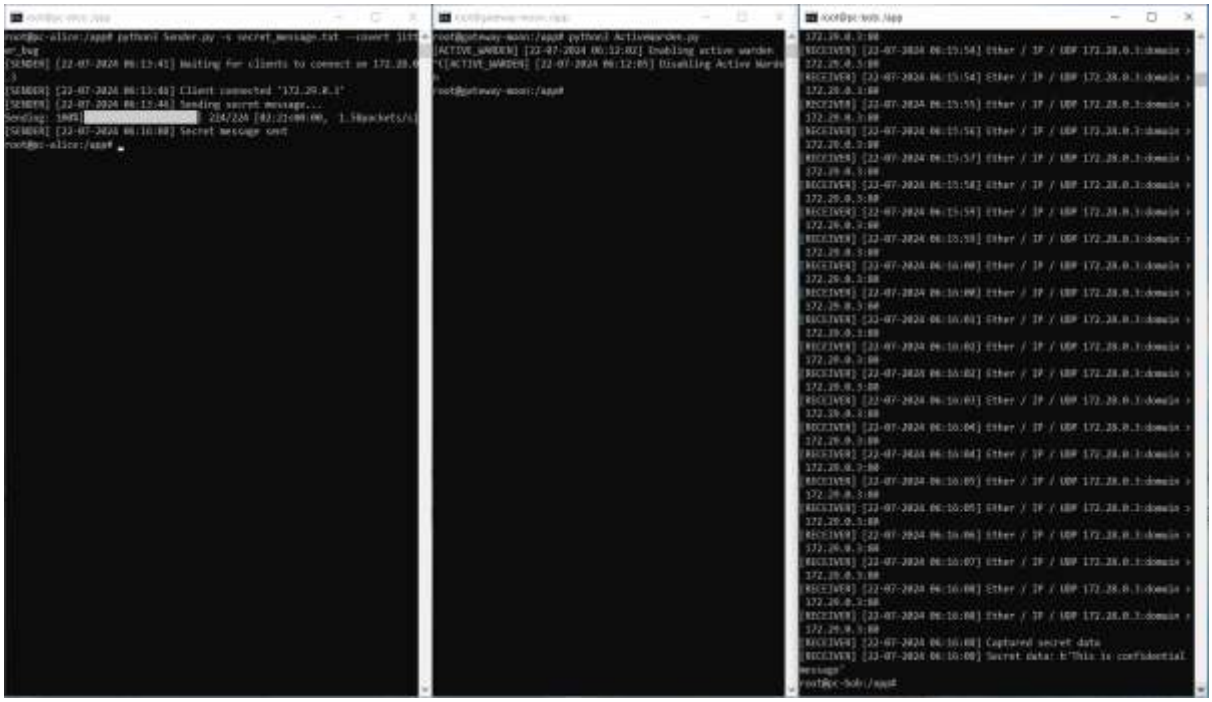

Figure 1. *Sender and receiver covert communication using Jitterbug covert Channel before enabling the active warden*

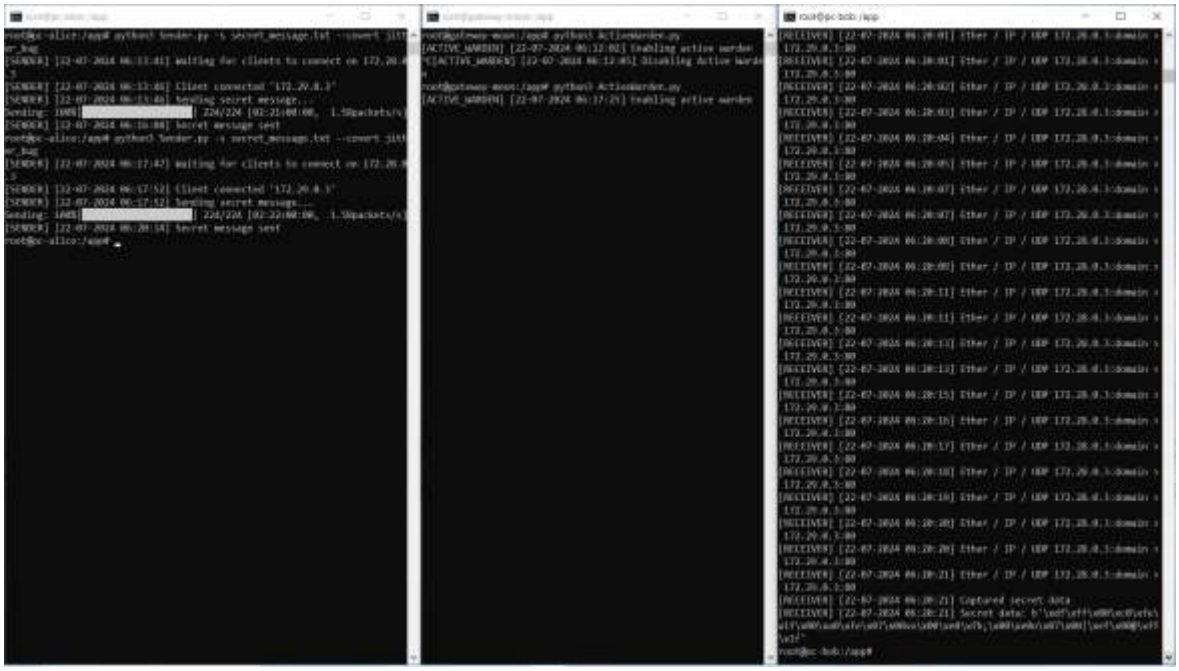

Figure 2. *Covert communication between Alice and Bob with Active Warden is enabled using Jitterbug covert channel*

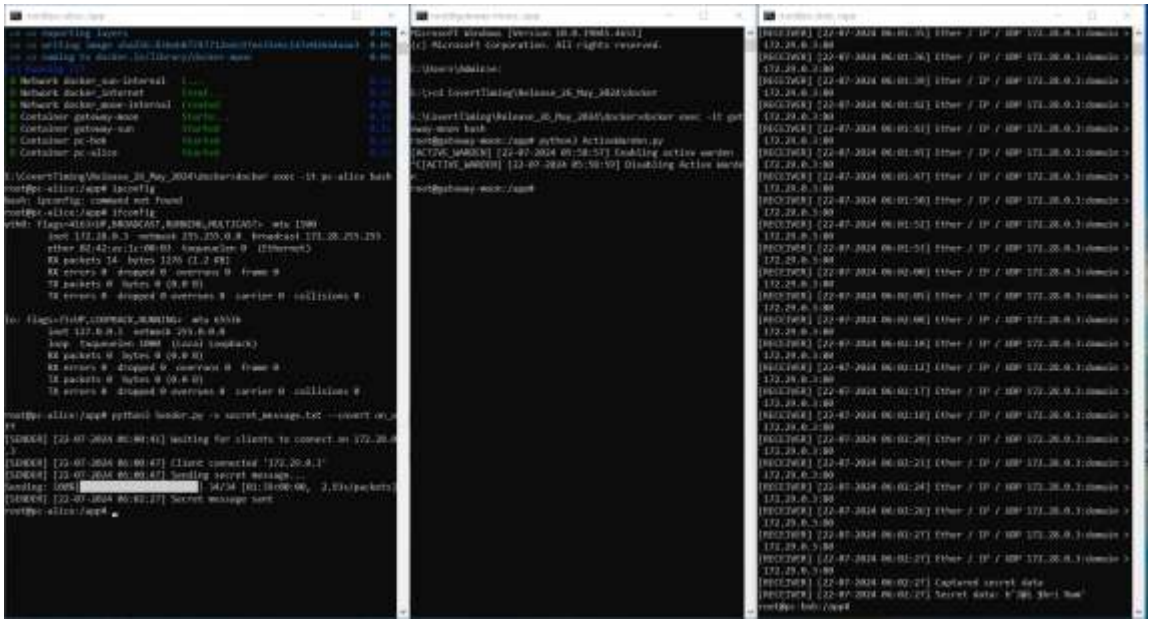

Figure 3. *Sender and receiver covert communication using On-Off encoding scheme*

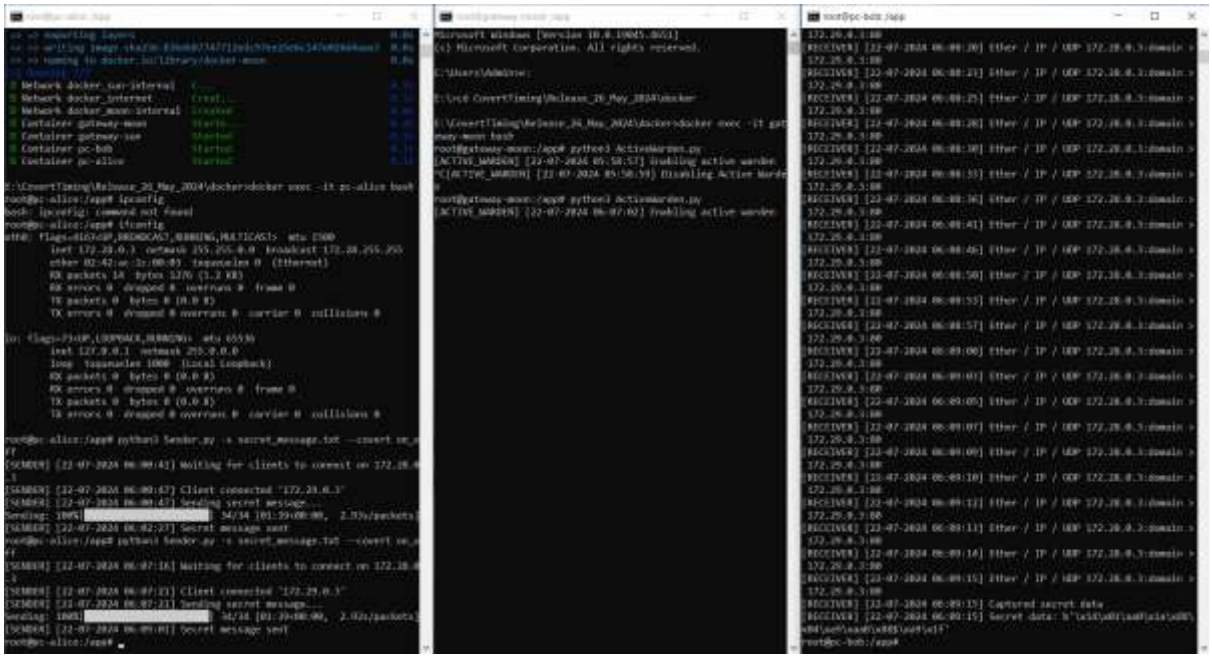

Figure 4. *Covert communication between Alice and Bob using On-Off scheme with Active Warden is enabled*

The sender sends the covert message to the receiver using the developed framework. On-Off Covert Channel is used as shown in Figure 3. In this, Sender started sending the covert message to the receiver using On-Off encoding scheme, the Receiver is signalling the sender to send message and it also shows the receiver has decoded the covert message sent by the sender using the On-Off decoding scheme.

In the figure 4 shows the covert communication between Alice and Bob using On-Off covert message encoding and decoding scheme before enabling the active warden and l_bits_to_n_packets Covert Channel is used as shown in Figure 5. Figure 6 shows the covert communication between Alice and Bob after enabling the active warden. It demonstrates that the active warden running on the gateway is enabled and normalizes the covert communication between Alice and Bob. The receiver actually does not receive the actual covert message, instead it decodes a corrupted message and hence the active warden has eliminated the covert communication.

The sender sends the covert message to the receiver using the developed framework. In this, Sender started sending the covert message to the receiver using On-Off encoding scheme, the Receiver is signalling the sender to send message and it also shows the receiver has decoded the covert message sent by the sender using the l_bits_to_n_packets decoding scheme. In the figure 5 as given above shows the covert communication between Alice

and Bob using l_bits_to_n_packets covert message encoding and decoding scheme before enabling the active warden and figure 6 shows the covert communication between Alice and Bob after enabling the active warden. It demonstrates that the active warden running on the gateway is enabled and normalizes the covert communication between Alice and Bob. The receiver actually does not receive the actual covert message, instead it decodes a corrupted message and hence the active warden has eliminated the covert communication.

2.2 Working of Active warden:

The ability to remove or block timing hidden routes was our primary goal while creating an active defence mechanism. This can be taken advantage of by changing the interarrival packet time or the delays in order to secretly communicate. The protocol normalization technique is employed by the suggested active warden defence model as shown in the figure 7 to eradicate potential covert channels.

A Traffic Interceptor system component that is essential for network monitoring and analytics. Its key features include:

a. Initiating Packet Sniffing: which involves capturing data packets as they move across a certain network. Packet sniffing allows you to intercept inbound and outgoing traffic for monitoring, debugging, or analysis. Wireshark, tcpdump, and proprietary systems are common examples of traffic interceptors.

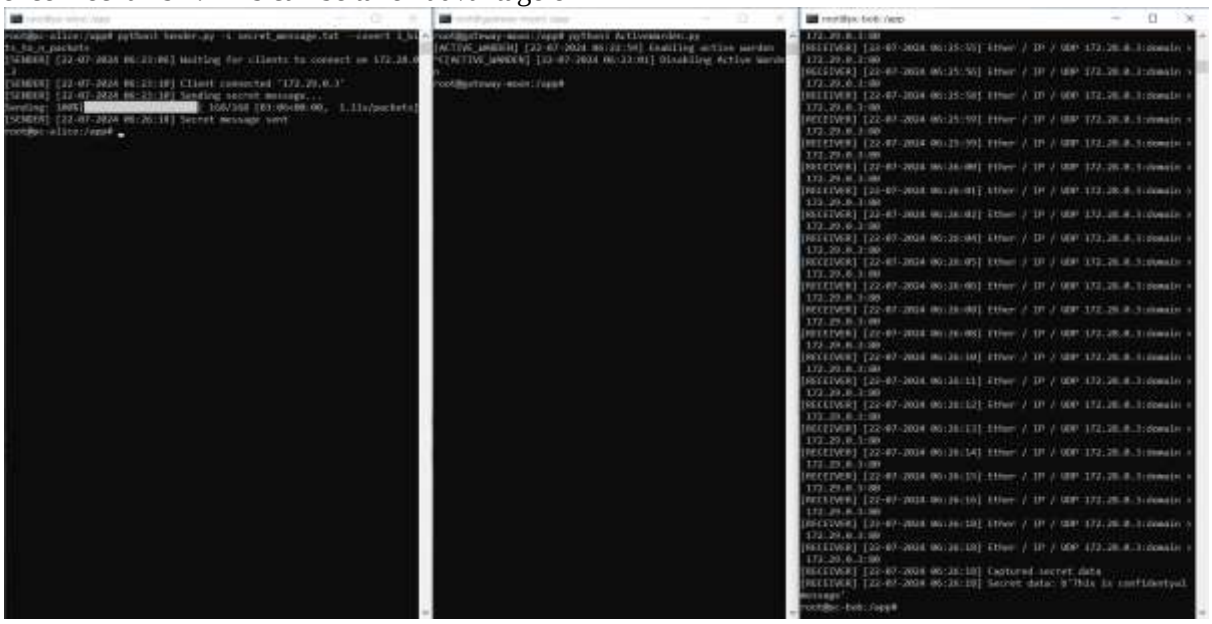

***Figure 5**. Sender and receiver covert communication using l_bits_to_n_packets encoding* scheme

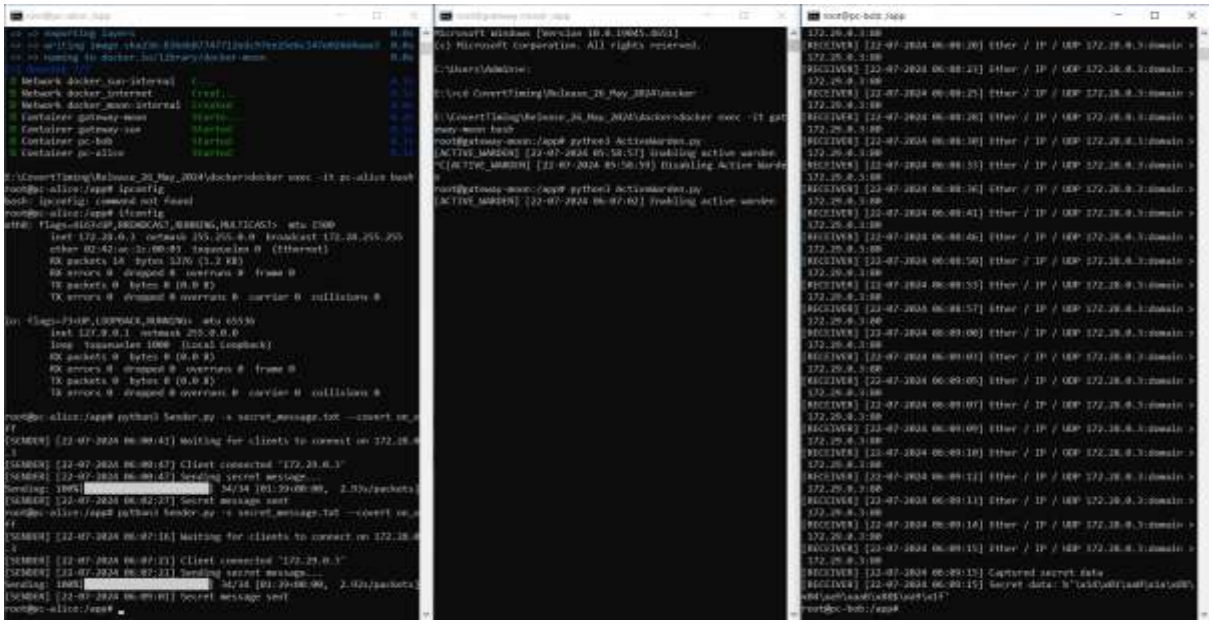

***Figure 6**. Covert communication between Alice and Bob using l_bits_to_n_packets scheme with Active Warden is enabled*

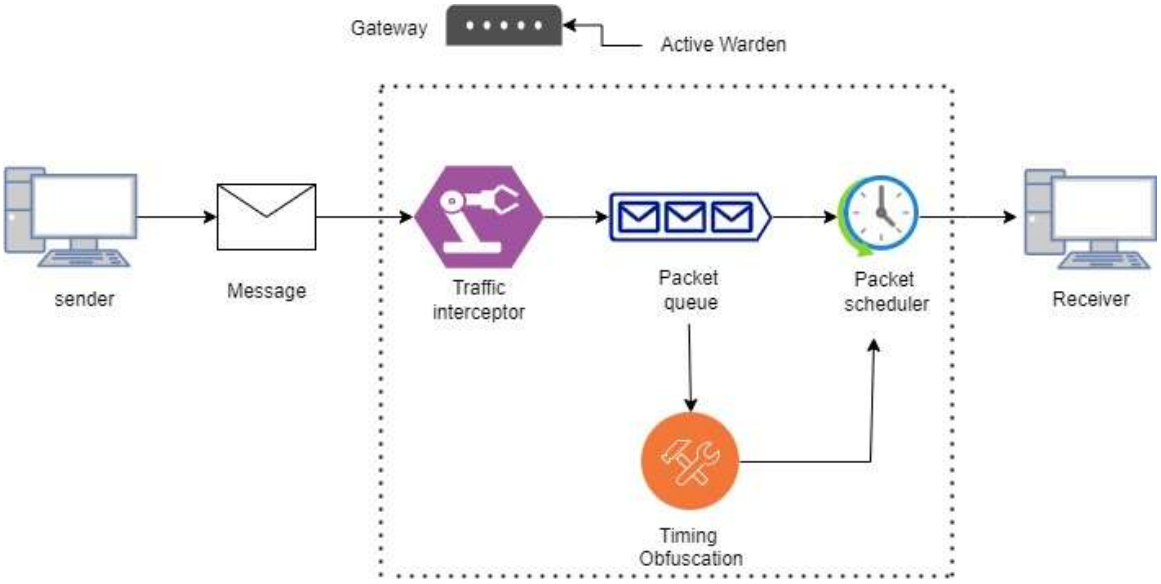

Figure 7. *Working of Active Warden*

b. Processing Network Traffic:

After packets are intercepted, the Traffic Interceptor processes them as they travel across the network which include the following:

Protocol Layers: Breaking down packet data into usable representations (for example, TCP headers and HTTP requests).

Data Filtering: The process of selecting only packets of interest based on certain criteria (such as IP addresses, ports, or protocols).

Logging and storing intercepted data for future study or archiving.

Real-Time Analysis: Providing information or alerts about potential problems such as abnormalities, invasions, or performance bottlenecks.

Package Queue

The packet queue temporarily stores captured packets from the network intercepted by a Traffic Interceptor for processing packets sequentially or in batches.

Delay List (Delays)

The delays list tracks time intervals between consecutive packet captures. The list illustrates the time difference between packet captures. This data can be used to evaluate network performance. Evaluate jitter, delay, throughput, support retransmission scheduling

The Packet Scheduler, schedules packet processing or forwarding based on computed delays or priority rules. Use the delays list to schedule packets for processing at specific intervals.

This is achieved by using the following steps.

Logging: The logging is essential for debugging and monitoring. Here, it indicates that the process of enabling the "active warden" has started.

Iptables utilty:

iptables is to set up, maintain, and inspect the tables of IP packet filter rules in the Linux kernel.

The -I option inserts a rule at the top of the specified chain (FORWARD in this case).

The -p tcp and -p udp specify the protocol (TCP and UDP respectively).

--dport {port} specifies the destination port.

-j DROP tells iptables to drop the packets that match the rule.

Threading:

Creating a new thread with threading. Thread and starting it allows the retransmit Packets method to run concurrently with the main thread. This is useful for performing tasks that can run independently. The retransmission of packets is done by observing the traffic flows, considering the recent delays. Retransmission is carried with the following steps:

Handling Overfull Queue:

In the Active Warden, we deploy a strong queue management system to monitor and regulate packet flow, allowing us to perform efficient normalisation of covert communication. During operation, the system determines if the packet queue size exceeds a maximum threshold that has been set in its configuration. The Active Warden will analyse the retransmission and retire the warden if the queue size falls below reasonable bounds.

To handle queued packets, the system employs a loop that keeps going until the packet queue is empty. In this loop, packets are collected one after the other from the queueing. The original packet format is reconstructed using the second and third elements in each packet tuple. Following normalisation, the packet is rebuilt, transmitted and then received without any issues.

We also remove the delays list for earlier retransmissions throughout this process. By resetting the timing settings and eliminating any potential residuals from previous delays, this procedure maintains a high level of system efficiency. The Active Warden may interfere with secret channels while maintaining the integrity of legitimate network traffic thanks to its comprehensive, active queue management technique.

Handling Non-Empty Queue:

If the queue size is non-zero but does not exceed the maximum threshold:

Generates a list of weights using a utility function. These weights are likely used to calculate the delay for retransmission.

$$w_i' = \frac{w_i}{\sum_{i=1}^{n} w_i}$$

Where w_i' is the normalised weight and the sum of the weights is given by the sum of the first n natural numbers

$$\sum_{i=1}^{n} = \sum_{i=1}^{n} j = \frac{n(n\ +\ 1)}{2}$$

Thus, the normalized weights are

$$w_i' = \frac{i}{\frac{n(n\ +\ 1)}{2}}$$

The delays are computed using a weighted quadratic mean of the delays list, and adds a random component in seconds for variability.

$$\text{weighted quadratic mean} = \sqrt{\frac{\sum_{i=1}^{n} w_i x_i^2}{\sum_{i=1}^{n} w_i}}$$

Retrieves a packet from the queue and removes the first element from the delays list, assuming it represents the delay associated with the oldest packet.Sends the packet using the sc.send method after reconstructing it.

Packet Capture:

The capture Packets method likely involves monitoring network traffic. This might be for security (like an IDS/IPS) or network management purposes.

Overall, Active warden acts a part of a network security or monitoring tool, where it sets up rules to block specific traffic, starts a process to handle retransmission of packets (potentially after inspection or modification), and begins capturing packets for analysis.

3. Results and Discussions

This section presents an efficacy analysis of the active warden, followed by the results indicating that the overhead associated with the active warden is small. The studies utilized different file sizes among five encoding schemes: Dual Bit, ON-OFF, JitterBug, L-Bits-to-n-packets, and Simple covert channel to ensure comparability. The duration for end-to-end (E2E) transmission was assessed with and without the Active Warden activated, indicating a marginal increase in transmission times when the security feature was operational. Table 1 illustrates the performance overhead introduced by the Active Warden mechanism during covert communication between a sender and receiver using different encoding schemes. The experiments were conducted with a consistent file size of 256 bytes to maintain comparability across five encoding schemes: Dual Bit, ON-OFF, JitterBug, L-Bits-to-n-packets, and Simple. The time taken for end-to-end (E2E) transmission was measured both with and without the Active Warden enabled, revealing a slight increase in transmission times when the security mechanism was active. The results show that, without the Active Warden, transmission times varied depending on the encoding scheme, with Dual Bit requiring 1877 seconds and JitterBug completing the transfer in 1337 seconds, reflecting differences in encoding complexities.

Table 1. *Overhead to Active Warden*

Covert Channel	File size (Bytes)	Time taken E2E-Sender to receiver - Active warden – Disabled (in Secs)	Time taken E2E-Sender to receiver- Active warden –Enabled (in Secs)	Overhead
Dual Bit	256	1877	1890	0.67
ON-OFF		2162	2174	0.57
JitterBug		1337	1348	0.83
L-Bits-to-n-packets		1707	1719	0.72
Simple		1835	1849	0.78

Table 2. *Overhead to Active Warden (512 bytes)*

Type of Covert Channel	File size (Bytes)	Time taken E2E-Sender to receiverActive warden – Disabled (in Secs)	Time taken E2E-Sender to receiverActive warden -Enabled (in Secs)	Overhead
Dual Bit	512	3755	3767	0.33
ON-OFF		4324	4336	0.29
JitterBug		2674	2685	0.42
L-Bits-to-n-packets		3413	3426	0.37
Simple		3669	3682	0.35

When the Active Warden was enabled, transmission times increased marginally due to the normalization processes it employed, such as delay randomization and protocol adjustments. For instance, the time for Dual Bit rose to 1890 seconds, while JitterBug increased to 1348 seconds. The overhead, calculated as a percentage increase in transmission time, remained minimal across all schemes, demonstrating the efficiency of the Active Warden. The ON-OFF scheme exhibited the lowest overhead at 0.57%, indicating minimal disruption to its communication. Conversely, the JitterBug scheme experienced the highest overhead at 0.83%, suggesting that its characteristics were more susceptible to the Active Warden's normalization techniques.

Table 2 presents the performance overhead associated with the Active Warden mechanism for covert communication involving a larger file size of 512 bytes. Similar to the previous analysis, the table evaluates five covert channel encoding schemes: Dual Bit, ON-OFF, JitterBug, L-Bits-to-n-packets, and Simple. The end-to-end (E2E) transmission times were measured with and without the Active Warden enabled to assess the impact of the security mechanism. The data indicates that transmission times increased slightly when the Active Warden was operational, reflecting the additional processing involved in normalization techniques. For instance, Dual Bit encoding required 3755 seconds without the Active Warden, compared to 3767 seconds when the mechanism was enabled. Similarly, the JitterBug scheme recorded an increase from 2674 seconds to 2685 seconds. These minor variations highlight the Active Warden's ability to enhance security without causing significant disruption. The overhead, expressed as a percentage increase in transmission time, was minimal across all encoding schemes. The ON-OFF scheme demonstrated the lowest overhead at 0.29%, indicating its robustness against the normalization processes. In contrast, JitterBug exhibited the highest overhead at 0.42%, suggesting that it incurred slightly more delays due to the Active Warden's interventions. Table 3 presents the overhead to the Active Warden mechanism for covert communication using a file size of 1024 bytes. The table includes five covert channel encoding schemes: Dual Bit, ON-OFF, JitterBug, L-Bits-to-n-packets, and Simple. Similar to the previous tables, the end-to-end transmission times are measured with the Active Warden disabled and enabled, allowing for the assessment of the overhead caused by the security mechanism.

The data reveals that, with the Active Warden enabled, transmission times increased marginally across all encoding schemes. For instance, the Dual Bit scheme required 7509 seconds with the Active Warden disabled and 7522 seconds with it enabled,

Table 3. *Overhead to Active Warden (1024 bytes)*

Covert Channel	File size(Bytes)	Time taken E2E-Sender to receiver Active warden – Disabled (in Secs)	Time taken E2E-Sender to receiver - Active warden -Enabled (in Secs)	Overhead
Dual Bit	1024	7509	7522	0.17
ON-OFF		8647	8660	0.15
JitterBug		5348	5357	0.18
L-Bits-to-n-packets		6827	6840	0.20
Simple		7339	7351	0.17

resulting in an overhead of 0.17%. Similarly, the ON-OFF scheme exhibited a minor increase from 8647 seconds to 8660 seconds, translating to an overhead of 0.15%. JitterBug, L-Bits-to-n-packets,

and Simple encoding schemes showed comparable increases in transmission time, with overheads ranging from 0.17% to 0.20%.
Since the overhead values are comparable to those of a 512-byte file, the Active Warden only slightly slows down communication in general. Since there is very little cost difference in the encoding strategies, the Active Warden encodes almost identically in various secret communication techniques.

4. Conclusions

The effectiveness of network security depends critically on the ability to eliminate covert communication channels (CCCs), the study shows. These channels exploit the weakness of the standard protocol that was designed to perform the communication on the networks, and thus they enable unauthorised data exchange, which is problematic in terms of privacy and data protection. In our experiment we found how such threats can be countered by the development and implementation of an adaptive warden mechanism. Through the use of traffic normalisation and real-time monitoring, we propose a dynamic covert traffic mechanism that adapts to covert strategies and mitigates covert communication while incurring only minimal performance overhead. Validation of the adaptiveness of the warden has been demonstrated via experimental results on Dual Bit, ON-OFF, and the JitterBug encoding schemes. We find that normalisation processes e.g., delay randomisation and changes to the protocol level that disrupt covert communications do this without significant impact on legitimate traffic. That the mechanism efficiently increases security while working to keep the network secure is proven by the constant less than 1 percent overhead on transmission times.
Nevertheless, the research points towards possible further investigative directions, especially in relation to covert channels within IPv6 contexts, which deserve to be better explored as they gain significance. Future work involves refining the adaptive mechanisms in response to the complexities of new protocols, as well as making scalability possible for large-scale network deployment. Finally, this work provides an important framework for studying and overcoming covert communication channels. Adaptive warden strategies can be integrated into network defences to increase organisations security posture and protect sensitive information against ever more sophisticated cyber threats.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] B. W. Lampson, (1973). A note on the confinement problem,"*Commun. ACM,* 16(10);613–615.
[2] S. J. Murdoch, (2007). Covert channel vulnerabilities in anonymitysystems, *Ph.D. dissertation, University of Cambridge.*
[3] C. H. Rowland, (2012). Covert channels in the TCP/IPprotocol suite, *First Monday*, vol. 2, no. 5, May1997, retrieved: Mar, 2012. [Online]. Available:http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/ar-ticle/view/528/449
[4] T. G. Handel and M. T. Sandford, II., (1996). Hiding data in the osinetwork model, in Proc. *First Int. Workshop on Information Hiding. London, UK:* Springer-Verlag, 1996, pp. 23–38.
[5] S. Cabuk, C. E. Brodley, and C. Shields, (2004). IP covert timing channels: design and detection," in *ACM Conference on Computer and Communications Security, V. Atluri, B. Pfitzmann,and P. D. McDaniel, Eds. ACM*, pp. 178–187.
[6] M. H. Kang, I. S. Moskowitz, and S. Chincheck, (2005). The pump:A decade of covert fun *ACSAC*, pp. 352–360.
[7] P. A. Porras and R. A. Kemmerer, (1991). Covert flow trees: A tech-nique for identifying and analyzing covert storage channels, *IEEE Symp. on Security and Privacy*, pp. 36–51.
[8] R. A. Kemmerer, (1983). Shared resource matrix methodology: anapproach to identifying storage and timing channels, *ACMTrans. Comput. Syst.*, 1(3)256–277.
[9] J. McHugh, (2001). An information flow tool for gypsy - an extendedabstract revisited, in *Proc. 17th Annual Computer SecurityApplications Conference*, pp. 191–201.

[10] C. Kr ¨atzer and J. Dittmann, (2006) Fr¨uherkennung von verdecktenKan¨alen in VoIP-Kommunikation, in *IT-Fr ¨uhwarnsysteme,ser. BSI-Workshop*. BSI, pp. 209–214, (In German).

[11] M. Handley, V. Paxson, and C. Kreibich, (2001). Network intru-sion detection: Evasion, traffic normalization, and end-to-endprotocol semantics, *in 10th USENIX Security Symposium*. 10, pp. 115–131.

[12] A. Singh, O. Nordstr ¨om, A. L. M. dos Santos, and C. Lu, (2006). Stateless model for the prevention of malicious communi-cation channels," *Int. Journal of Comp. and Applications,*28(3);285–297.

[13] G. Gu, R. Perdisci, J. Zhang, and W. Lee, (2008). Botminer: Clus-tering analysis of network traffic for protocol- and structure-independent botnet detection, *USENIX Security Symp.*, pp. 139–154.

[14] S. Zander, G. Armitage, and P. Branch, "Covert channelsand countermeasures in computer network protocols," *IEEEComm. Magazine,* 45(12),136–142.

[15] Daemon9, (2012). Loki2 (the implementation)," Phrack Magazine, 7(5); September 1997, retrieved: Mar, 2012. [Online].Available: http://www.phrack.org/issues.html?issue=51&id=6

[16] S. Wendzel and J. Keller, (2011). Low-attention forwarding formobile network covert channels, in *12th IFIP Comm. andMultim. Security, ser. LNCS,* . 7025;122–133.

[17]S. Wendzel, (2008). Protocol hopping covert channels, *Hakin* 08(03);20–21, 2008, (in German).

[18] Steffen Wendzel Protocol channels as a new design alternative of covertchannels, *CoRR*, abs/0809.1949, pp. 1–2, 2008.

[19] Steffen Wendzel (2011). Analyse der Pr ¨aventions- und Detektionsmethoden f¨urverdeckte Kan¨ale," *Master's thesis, Augsburg University Applied Sciences,* (in German).

[20] C.-R. Tsai and V. D. Gligor, (1988). A bandwidth computation modelfor covert storage channels and its applications," in *Proc.IEEE Conf. on Security and Privacy* pp. 108–121.

[21] S. Wendzel, "pct," 2009, retrieved: Mar, 2012. [Online]. http://www.wendzel.de/dr.org/files/Projects/pct/

[22]D. Berrange, "Simulating WAN network delay,"2005, retrieved: Mar, 2012. [Online]. http://people.redhat.com/berrange/notes/network-delay.html

[23] J. Morris, (2012). IPTables::IPv4::IPQueue module for Perl,"2002, retrieved: Mar, 2012. [Online],http://search.cpan.org/˜jmorris/perlipq-1.25/IPQueue.pm

[24] C. D. Mee and E. D. Daniel, Magnetic Storage Handbook,2nd ed. McGraw Hill, 1996.

[25] T. Kohno, A. Broido, and k. claffy, (2005). Remote physical devicefingerprinting, *IEEE Transactions on Dependable and Secure Computing,* 2;93–108.

[26]Akamai, (2012). Retail web site performance, retrieved: Mar, 2012. [Online]. http://www.-akamai.com/dl/reports/Site Abandonment Final Report.pdf

[27] Uttarwar Vrushali Uday, Dhananjay M. Dakhane, and Khushi P. Sindhi. *Novel Framework for Evaluating Covert Channels and Its Countermeasures in Network Protocols.* Published 4 June 2024.

[28] M. Chourib, S. Wendzel and W. Mazurczyk, (2021). Adaptive Warden Strategy for Countering Network Covert Storage Channels, 2021 *IEEE 46th Conference on Local Computer Networks (LCN*), Edmonton, AB, Canada, 2021, pp. 148-153, doi: 10.1109/LCN52139.2021.9524939.

[29]Adams, L., & Green, M. (2021). *Dynamic normalization techniques for adaptive wardens in covert communication systems*. *Journal of Network Security*, 15(2), 123-139.

[30]Brown, H., et al. (2020). Enhancing covert communication: The role of adaptive wardens and rule-based normalization. *International Journal of Cyber Security*, 22(4), 199-214.

[31] Cheng, Y., & Liu, F. (2020). Leveraging Libpcap and Scapy for covert communication in modern network systems. *Proceedings of the 2020 International Conference on Network Security*, 82-90.

[32] Doe, J., & Miller, A. (2019). Adaptive warden systems in covert communication networks: A comprehensive review. *Network and Information Security Review*, 10(1), 50-66.

[33] Kumar, S., et al. (2018). Randomized rule activation for covert channels in adaptive warden systems. *Journal of Communications and Network Security*, 9(3), 200-215.

[34] Lee, T., & Zhang, J. (2017). Evaluating the effectiveness of dynamic wardens in concealing covert communications. *Journal of Information Security and Privacy*, 34(5), 154-170.

[35] Sharma, R., et al. (2020). Normalizing network traffic for covert communications using Netfilter: Techniques and challenges. *International Journal of Network Security and Privacy*, 11(2), 100-115.

[36] Singh, R., & Patel, S. (2022). Performance evaluation and optimization of covert communication systems with adaptive wardens. *Journal of Cybersecurity Research*, 14(4), 305-318.

[37]Smith, D., & Jones, M. (2018). Covert communication using adaptive wardens: A network-centric approach. *International Journal of Covert Networks*, 5(1), 72-88.

[38] Wang, Z., et al. (2019). Covert communication protocols and data hiding techniques in packet-based networks. *Journal of Network Protocols*, 17(3), 140-155.

[39] Zhao, Q., & Liu, W. (2021). Covert communication through adaptive wardens: A performance and feedback evaluation. *International Conference on Information Security and Networks*, 189-198.