applied sciences

MDPI

Article

Detecting Malicious Devices in IPSEC Traffic with IPv4 Steganography

Gabriel Jekateryńczuk [1,*], Damian Jankowski [1], René Veyland [2] and Zbigniew Piotrowski [1]

[1] Faculty of Electronics, Military University of Technology, 00-908 Warsaw, Poland; damian.jankowski@wat.edu.pl (D.J.); zbigniew.piotrowski@wat.edu.pl (Z.P.)
[2] Thales, 4, Avenue des Louvresses, 92622 Gennevilliers Cedex, France; rene.veyland@thalesgroup.com
* Correspondence: gabriel.jekaterynczuk@wat.edu.pl

Abstract: This study investigates the application of steganography for enhancing network security by detecting and promptly eliminating malicious packets to prevent flooding and consequent denial of service attacks while also identifying malicious equipment. The paper discusses foundational concepts such as the prisoner's dilemma, covert channels, qualitative metrics, and existing steganography techniques in computer communications. An architecture was developed to assess the effectiveness of this solution, and experiments were conducted, with their results presented. This contribution leverages established steganographic principles and seamlessly integrates with widely adopted IPsec protocols, offering a solution to improve covert communication within computer networks.

Keywords: steganography; network security; data privacy; malicious traffic; network monitoring

check for updates

Citation: Jekateryńczuk, G.; Jankowski, D.; Veyland, R.; Piotrowski, Z. Detecting Malicious Devices in IPSEC Traffic with IPv4 Steganography. *Appl. Sci.* **2024**, *14*, 3934. https://doi.org/10.3390/app14093934

Academic Editors: Masoud Kaveh and Diego Martín

Received: 26 March 2024
Revised: 24 April 2024
Accepted: 29 April 2024
Published: 5 May 2024

1. Introduction

In contemporary society, the evolution of steganography is intricately linked to the advancements in information technology. Consequently, the emergence of novel forms of interpersonal communication, such as chat, video, and audio, gives rise to innovative data "carriers" and, consequently, modern steganographic techniques.

Steganography's core principle is to discreetly convey information, ensuring that external observers remain oblivious to the information's presence and the establishment of covert communication. This sets steganography apart from cryptography, which aims to safeguard transmitted content from unauthorized access, potentially revealing the act of communication itself. This term is often confused with other solutions, such as watermarking [1,2] and cryptography [3] (Figure 1). Nevertheless, it is crucial to distinguish steganography from these other approaches. Cryptography primarily focuses on safeguarding the content of messages, ensuring the confidentiality and integrity of their meaning. On the other hand, watermarking is dedicated to establishing information identity, thereby deterring unauthorized usage.

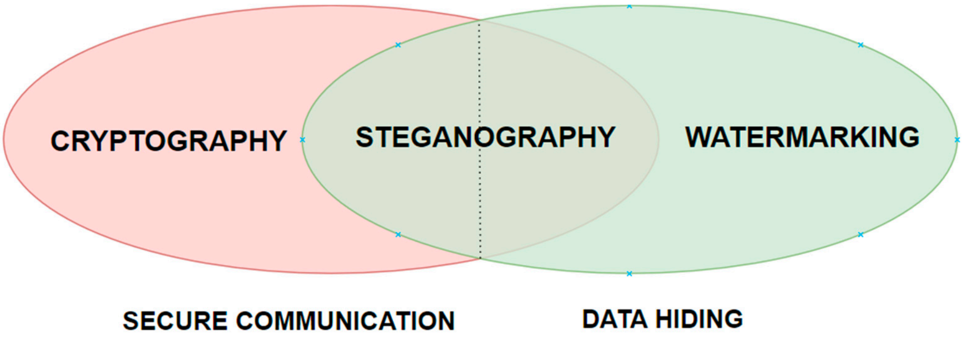


Figure 1. Relationship between cryptography, steganography, and watermarking [4].

Steganographic methods have been recognized and employed for centuries. The term "steganography" originates from Greek and translates to hidden writing. The earliest steganographic techniques can be traced back to the fifth century BCE. Herodotus documented the transmission of information regarding planned attacks on Greece, describing the concealment of messages within the abdominal cavities of hunted animals or through tattooing messages on the heads of trusted slaves, sending these covert messages once their hair regrows. Other known steganographic techniques used over the centuries include using sympathetic ink and micro-dots, placing hidden information in sheet music or plain text, and then reading such hidden data using an appropriate template on the text [5].

Over time, with the rapid evolution of computer technology, a wide range of techniques for hiding data have been introduced into steganography. Remarkably, steganographic techniques have found favor with images [5], allowing data to be discreetly included in an image's pixel structure without compromising the image's perceived quality. Audio [6] has also emerged as a significant information carrier in steganography. Many ways to conceal data, including an audio signal's frequency and amplitude components, have been tested. This medium adds another degree of camouflage because some frequencies cannot be heard by the human auditory system [6]. Moreover, steganography in videos [7] has been made possible by developing computational powers and its extensive data capacity. This medium allows for covertly embedding information within visual content and extends the potential for concealing data within the audio components. Additionally, the recent rapid advancement of artificial intelligence development has increased its popularity because new techniques are often more accurate than conventional steganography techniques and allow overcoming limitations, e.g., compression and robustness [8]. Another approach is coverless steganography, which achieves covert communication without modifying the carrier itself. Instead of embedding data directly, coverless steganography relies on mapping the concealed information and the carrier medium, whether an image, video, or audio [9–11]. Beyond multimedia, steganography has extended its reach to internet protocols [12], offering solutions to various network security issues. By embedding information within data packets transmitted over the network and manipulating bits within these packets or their payloads, steganography enables covert data transmission across the internet without raising suspicion. This technique addresses data exfiltration, covert communication, and unauthorized access detection within network security frameworks. Moreover, steganography's utility extends to practical scenarios such as secure data transmission, hiding encryption keys, watermarking sensitive information, and detecting malicious activities such as covert channels or stealthy communication, enhancing network systems' overall security posture [13–16].

The aim of this paper is to develop a steganographic procedure that enhances network security by monitoring traffic and identifying malicious devices in order to prevent flooding and DDoS attacks. Our approach also describes the interception process of MAC addresses associated with these devices. Furthermore, we introduce a service that enables efficient monitoring of network packets within a network using the visualization platform Grafana.

The paper is organized as follows: Section 2 contains a description and overview of the essential concepts in network steganography and existing methods. Section 3 presents a novel concept of enhancing network security with steganography used in IPv4 over IPsec. Section 4 delves into the architecture employed in the study, while Section 5 presents the results and their analysis. Section 6 encompasses the conclusions drawn from the study's findings.

2. Network Steganography

At a time when image steganography was the most used and recognizable form of steganography, a new type of steganography, using network protocols as information carriers, was gaining popularity. Network steganography is a generic term that may describe all information concealment techniques used to communicate in telecommunication networks. This nomenclature was proposed for the first time in [17]. The author introduced a novel network steganography method named the Hidden Communication System for Corrupted Networks (HICCUPS) for wireless local area networks (WLANs). Since then, steganography has developed into an essential field of study for data concealment and safe communication in computer communications.

2.1. Prisoner Dilemma

The prisoner's dilemma was first described by G. J. Simmons [18]. It is an anecdote that perfectly describes the essence of steganography—an attempt to carry out a hidden exchange of information. G. J. Simmons described the story of two prisoners held in separate and non-adjacent cells who would agree on a detailed plan to escape from the prison and warn each other of the threats posed by the guards. Their interactions are characterized by unstable communication and constant observation by guards assigned to report unusual behavior among prisoners. Prison communication would be difficult if cryptography were used to exchange information since the guards would view it as a conspiracy. Hence, cryptography is not a good way to secure messages exchanged by prisoners. Therefore, prisoners must use a method of conveying information that conceals this communication's existence—steganography. The entire communication process is illustrated in Figure 2.

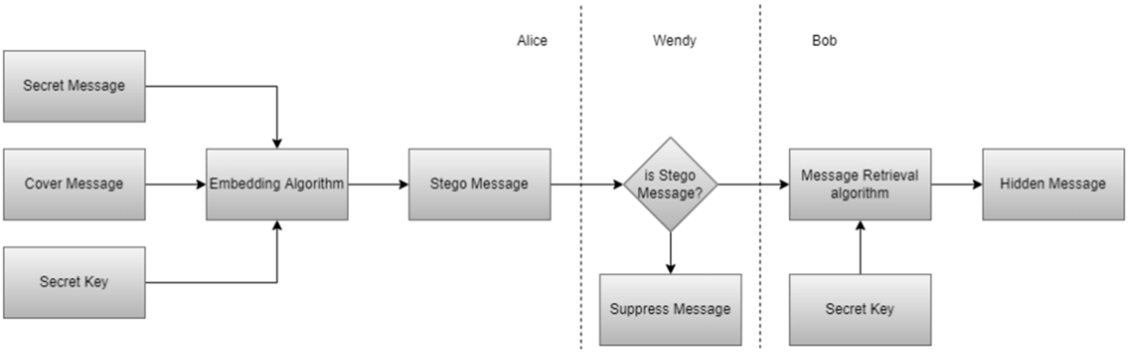


Figure 2. Prisoner's dilemma.

Before the arrest, prisoners have a message authentication scheme, which is done by entering redundant information by the sender. In such conditions, their communication must run through the subliminal channel, which enables the establishment of hidden communication between the sender and the recipient of an inconspicuous-looking message [19]. In practice, the prisoners are the sender and recipient of the message, while the guard is a network node that inspects the message using the deep packet inspection (DPI) functionality.

In the communication channel, attempts to detect and counteract hidden communication are called steganalysis. It is usually carried out using the so-called wardens—network nodes equipped with deep packet inspection functionality. Craver [20] describes the following behavior of the wardens:

- Passive behavior—consists of monitoring messages sent through the communication channel without introducing changes to the message.
- Active behavior—consists of introducing changes to the messages transmitted in a way that does not modify the message's meaning or disrupt or limit communication between nodes that do not use steganographic methods.
- Malicious behavior—like active behavior, with the difference that counteracting hidden communication may lead to interruption, disruption, or restriction of communication between nodes that do not use steganographic methods.

Additionally, there are three distinct countermeasures against covert channels [21]:

- Functionality—Wardens counteract covert channels either passively by capturing traffic or actively by manipulating packets, with options for stateless or stateful processing and dynamic adaptation.
- Knowledge—Wardens use various types of information to detect covert channels, ranging from network awareness to being agnostic about the channel's characteristics. They might focus on particular patterns or threats or have pre-existing knowledge regarding how data is hidden.
- Localization—Wardens are deployed at specific locations within a network, ranging from local devices to cover large areas or adopting a distributed approach across multiple nodes. Modern wardens must scale to protect large network trunks and cooperate across replicas to detect threats effectively.

In addition to the behaviors outlined by Craver, it is crucial to consider the active interference that wardens, such as Wendy, could engage in within the communication channel. One such scenario involves Wendy substituting messages exchanged by the prisoners. By intercepting and altering the messages, Wendy could manipulate the flow of communication, potentially undermining the prisoners' plans or causing confusion, thus preventing the creation of a covert channel. Furthermore, Wendy could also fabricate messages that deceive the prisoners into believing they are communicating with each other when, in reality, they are interacting with Wendy.

2.2. Covert Channels

"Any communication channel that a process can exploit to transfer information in a manner that violates the system's security policy" [22] is how the US Department of Defense defines the term "covert channel." In other words, a secret channel is a method of information hiding whereby a user transfers confidential data between two or more entities using a regular communication channel, all while keeping the information hidden from outside parties. The general idea of the covert channels in information and communication technology (ICT) networks is illustrated in Figure 3.

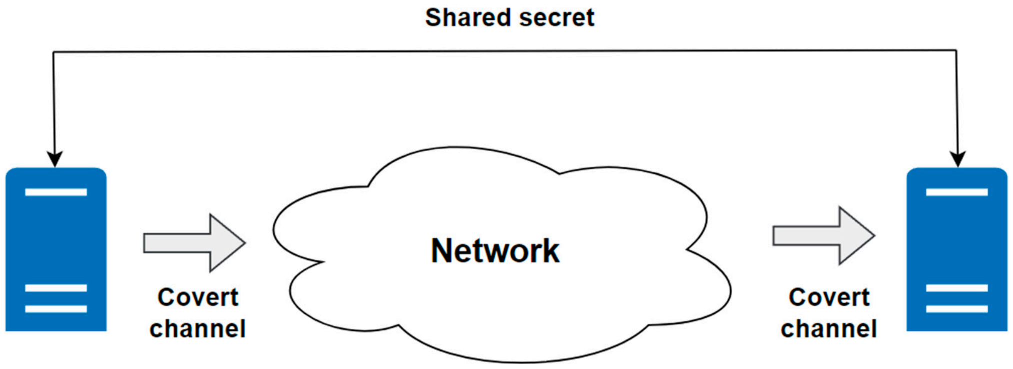


Figure 3. Covert channels in ICT networks.

A covert channel in ICT networks can be established by utilizing network steganography between network devices. While both end devices are aware of the secret message, the other devices within the network remain unaware of its existence. There are two fundamental types of covert channels [23]:

- Storage Covert Channel—it uses the fields of the network packet as the message carrier to transmit the steganography message. Headers are crucial because they include the data needed to start and continue a communication. While some fields are necessary for proper flow, others are optional and can be modified to hold covert data.
- Timing Covert Channel—it implements communication by modifying the behavior of the transmission system by intentionally changing the time dependencies between transmitted packets. One drawback of approaches employing this functionality is the requirement for a synchronization mechanism [24].

Other than the mentioned hidden channels, hybrid solutions combine the above two cases.

The transmission of information using the storage covert channel mechanism may be performed using one or more protocols. When using one protocol, the data is hidden in the packet structure. On the other hand, when using different protocols, the data need not be encoded in the packet, but the packet itself can be interpreted as a message. The use of multiple channels is more effective when sending short messages, as the channels created in this way are not efficient in terms of bandwidth compared to the storage channel, which uses fewer network protocols.

In our novel approach, a storage covert channel has been used.

2.3. Steganography for OSI RM Model

The Open Systems Interconnection Reference Model (OSI RM) was created to standardize how information flows over IP networks [25]. It separated the communication process into seven layers. The network steganography technique allows the construction of concealed communication channels and covert communication between network users in each of the OSI RM model's layers. Table 1 presents an overview of existing methods in the literature.

Table 1. Network steganography methods in literature.

OSI Layer	Name	Medium/Protocol	Capacity	Reference Number
Physical	WiPad	OFDM	1.54 Mbit/s	[26]
	-	PHY frame	1 bit/packet	[27]
	-	DFT-precoded OFDM	-	[28]
Data link	HICCUPS		44–216 kbit/s	[17]
	-	MAC frames	1–22 bytes/frame	[27]
Network	-	IPv4	max 4 bytes/packet	[29–33]
	-	IPv6	max 140 bits/packet	[34–36]
	ICMPStegano	ICMP	max 60 bytes/packet	[37]
	ARPNetSteg	ARP	44 bits/packet	[38]
	PadSteg	ARP + TCP	32 bit/s	[39]
Transport	RSTEG	TCP	~max 1.4 kB/packet	[40]
	-	TCP	30 bits/packet	[41,42]
	-	UDP	4 bits/packet	[43]
	-	SCTP	10–500 bits/s	[44]
Session	-	SIP + SDP	2.36 kbit/VoIP call	[45]
	-	SIP	464–1856 kbit/s	[46]
Presentation	-	-	-	[47]
Application	LACK	VoIP (RTP)	max 590 bits/s	[48]
	TranSteg	VoIP (RTP)	~32 kbit/s	[49]
	-	HTTP	2–2^{30} bit/connection	[50]
	-	SSH	12 bytes/packet	[51]

- Physical Layer—compared to the higher layers of the OSI RM model, covert channels in the lower layers are more challenging to execute and, hence, harder to find. One of the network steganography methods that uses the communication channel's physical characteristics is the wireless padding (WiPad) method for wireless local area networks (WLANs). This method consists of filling bits into the padding of transmission symbols. Another solution has been proposed, which is using DFT-precoded OFDM. The authors suggested sending a cover signal in addition to the weak secret signal rather than delivering the weak signal to the authorized recipient. In [27], the author proposed using the PHY Service Data Unit field, where seven of eight bits are used. The data can be hidden because one bit is reserved and not used.

- Data Link Layer—steganography at the data link layer can be implemented using methods based on the use of transmission frames with intentionally erroneous checksums. The HICCUPS method, which uses the mentioned mechanism, was proposed. In ICT networks, frames with erroneous checksums are rejected by physical drivers (network cards), while cards aware of using steganographic methods can read such packets and extract secret data from the frame. An interesting solution was proposed, which was to hide data in MAC frames. We can achieve different capacities based on chosen frames: the data frame, the beacon frame, the acknowledgement frame, and the MAC command frame. In [52], the authors proposed a network access point authentication mechanism that leverages beacon frames and timestamp fields, exploiting the least significant bits of these fields.
- Network Layer—the most popular protocol to hide data in the network layer is IPv4. The authors propose several methods based on hiding data in creating covert channels, such as hiding data in the identification field, overflowing the timestamp option field, and using time-to-live or flags. Murdoch [33] summarized the possibilities of using individual header fields of the IPv4 protocol. Another interesting approach is steganography with IPv6. This technique leverages specific fields within the protocol, namely the traffic class, flow label, and source IPv6 address. An ICMP-based approach has been suggested; however, it has not gained as much popularity as other protocols, possibly due to its limited number of header fields. Additionally, methods utilizing ARP have been developed. However, ARPNetSteg, a technique employing covert communication through ARP spoofing, can also be implemented in the data link layer, as it does not strictly belong to either layer. The method involves synchronized generation of lists of unallocated local IP addresses using a shared seed value, enabling sender and receiver devices to exchange covert data by manipulating ARP requests and replies, with embedded control information indicating the sender’s intent to transmit additional covert data. An alternative approach is the PadSteg method, which employs both ARP and TCP protocols. In this technique, a steganogram is embedded into the padding of Ethernet frames, requiring examination of the respective layer protocols (ARP or TCP) to ascertain the presence of secret data.
- Transport Layer—the current networking advancements propose techniques for modifying TCP headers. Specifically, the fields open to modification include the reserved field and the urgent pointer field. Another interesting technique of network steganography in the TCP protocol is the retransmission steganography (RSTEG) method. The method rejects acknowledgement of successfully received segments to trigger retransmission intentionally. The retransmitted packet contains secret information hidden in the data field. The possible steganographic throughput per packet is about 1.4 kbit. The maximum possible bandwidth of the RSTEG method is even 8 kbit/s, with a targeted level of packet retransmission reaching 5%, where the average level of retransmission on the internet is 3–7%. Additionally, a method based on UDP has been suggested, wherein hidden data is transmitted within the length of the network packets. However, UDP introduces randomness in the packets since there is no ACK functionality like in TCP. Another usage is the stream control transmission protocol (SCTP), which was developed to replace the TCP protocol. The authors proposed methods that modify the content of packets and the way they are exchanged and hybridized for both scenarios.
- Session Layer—tokens and fields within the SIP protocol were suggested for concealing data. An alternative covert channel involves manipulating header order or combining upper and lower-case letters in SIP headers. The authors recommend establishing concealed channels for SIP and the closely associated SDP protocols, incorporating security measures to guarantee authentication and confidentiality for sensitive information. The authors attained a throughput of 10 to 500 bits per second in their experimental scenario. However, authors in [40] calculated the number of SIP

messages that can be sent without raising the alarm and achieved 464 to 1856 kbit/s based on the test scenario.

- Presentation Layer—the multimedia elements may function as covert data transport channels. Since multimedia files frequently include vast amounts of data, they make appealing hosts for secret data storage. In [47], the author presents examples of embedding concealed information within images, audio files, and texts. Various techniques, such as Least Significant Bit (LSB), phase coding, and echo data hiding, can be employed. It is important to note that these techniques are not exclusive to network steganography. Instead, they pertain to steganography across visual, audio, and text mediums, with the concealed data subsequently transmitted over a network for decoding the steganogram on the recipient's end.
- Application Layer—this layer is the uppermost and is responsible for the user interface. One of the most popular and most used protocols is the Hypertext Transfer Protocol. Authors in [50] proposed nine new covert channels. For this purpose, authors use a protocol feature that has a dual nature, meaning the same feature can be obtained in more than one way, the feature is not mandatory, there exists a random value field, or there is no strict rule on how to obtain new values for some fields. Another interesting approach is called lost audio packets steganography (LACK), which uses the fact that excessively delayed real-time protocol (RTP) packets are always discarded and not used to reconstruct transmitted data. An even higher capacity named transcoding steganography (TranSteg) has been proposed. The key innovation of TranSteg lies in its ability to select a codec for a chosen voice stream that maintains similar voice quality but results in a smaller voice payload size than the original codec. The transcoding process occurs in a manner that intentionally preserves the original voice payload size, and the codec change is concealed. Subsequently, the transcoded voice payload is embedded, and any remaining free space is utilized for hiding data. An SSH-based method was also suggested, involving concealment by creating a MAC-like message and incorporating supplementary encrypted content into the packet.

There are undoubtedly other network steganography approaches, such as using protocols from many layers at once or modifying packet headers along with packet ordering, that can accomplish different bandwidths in addition to the ones mentioned above.

2.4. Qualitative Measures

All steganographic methods can be described using basic quality measures. They are often represented using a geometric figure, such as an equilateral triangle (Figure 4) [53].

This signifies that fundamental quality measures like bandwidth, robustness, and undetectability are interdependent. They can be described as follows:

- Bandwidth—this is the capacity of the steganography method. Bandwidth is frequently expressed in bits per second or bits per packet. It refers to the maximum data transfer capacity per time unit or one packet accordingly.
- Robustness—this is the feature that determines the degree to which a steganogram can withstand changes without compromising the integrity of the hidden data.
- Undetectability—this is defined as the inability to localize steganographic data in network protocol. Analysis of the statistical characteristics of network packets is frequently used to detect it.

Therefore, improving one of these metrics typically causes the others to deteriorate. For instance, sacrificing some of the available bandwidth can be necessary to increase the concealment of information. The ideal steganographic method should have high bandwidth and be robust and undetectable. However, creating practical and efficient steganography techniques depends on striking the ideal balance.

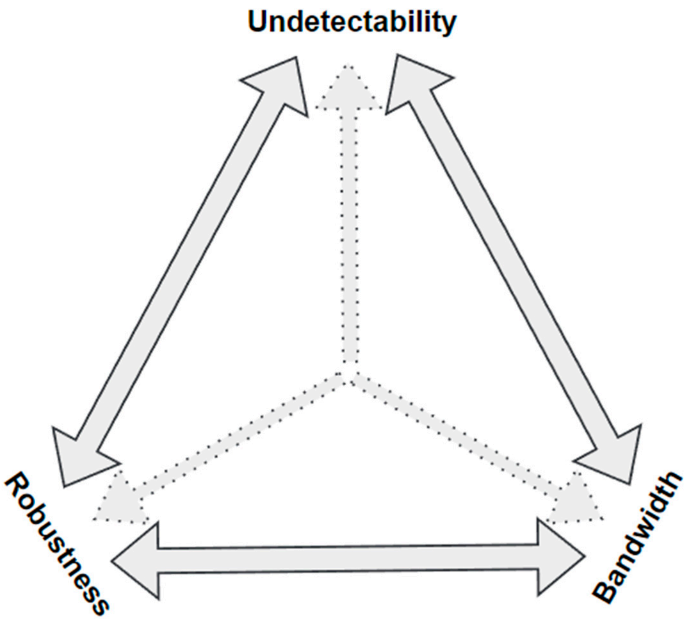


Figure 4. Relationships between qualitative measures.

3. IPsec-Based Network Steganography

The utilization of IPsec in steganography is a topic that has yet to be explored in the literature. So far, only one example of the covert channel with IPSec has been proposed in the literature [54]. This lack of exploration is particularly notable when considering the development of a concealed storage channel, primarily due to the constraints posed by the limited number of available header fields. Nevertheless, despite these challenges, and although IPsec is widely regarded as a very secure protocol, there is substantial potential to enhance network nodes' security and authorization capabilities by implementing steganography in the IPv4 protocol. Additionally, around 2015, the U.S.'s NSA reportedly discovered vulnerabilities in its security by breaching IPsec or tampering with Diffie-Hellman algorithm keys. Additionally, in 2018, a team of researchers found out that reusing a key pair across different versions and modes of IPsec IKE can lead to cross-protocol authentication bypasses, enabling the impersonation of a victim host or network by attackers. If the attack was completed, they could complete the handshake with the victim's device [55].

Our innovative approach introduces a novel method that enhances the integrity of transmitted packets on the IP header level and enables the potential to manage incoming traffic proactively. In detecting corrupted information, our method empowers network devices to initiate actions such as blocking incoming traffic. Furthermore, our approach facilitates communication with end hosts, alerting them to altered traffic within the packet's path by providing information about the MAC address of the corrupted network device.

3.1. IPsec Overview

The full name of IPsec is Internet Protocol Security. The "IP" part indicates the data's destination, while the "sec" part is responsible for their encryption and authentication. In other words, IPsec is a set of protocols that establish a secure and encrypted connection between devices on the public internet.

The protocols within the IPsec set are typically grouped according to their tasks [56]:

- Authentication Header (AH)—The AH protocol ensures data integrity, data origin authentication, and optional replay protection. It achieves this through a message digest generated by algorithms like HMAC-MD5 or HMAC-SHA and a shared secret key. Replay protection is implemented via a sequence number field in the AH header. AH authenticates IP headers and payloads, excluding specific transit-permissible fields like time-to-live (TTL).

- Encapsulating Security Payload (ESP)—The main idea of this protocol is to provide data confidentiality (encryption) and authentication (data integrity, data origin authentication, and replay protection). ESP can operate with confidentiality only, authentication only, or both. In authentication mode, ESP employs algorithms akin to AH but with distinct coverage. While AH-style authentication encompasses the entire IP packet, including the outer IP header, ESP's mechanism authenticates solely the IP datagram portion of the packet.
- Security Association (SA) manages establishing, maintaining, and terminating secure connections between network entities. It is crucial to define and agree upon security parameters to ensure a consistent and secure channel for data exchange.

IPsec also allows two modes of work:

- Transport mode—encrypts the outgoing data but does not conceal information about where it is intended to go. This means that a hacker may not be able to decipher intercepted messages but would know when and where they were sent.
- Tunnel mode—establishes a secure connection between two devices on the internet, creating a fully private link. In this mode, an IPsec-based virtual private network (VPN) operates as it forms a VPN tunnel.

Figure 5 illustrates the packet structures for AH and ESP in both transport and tunnel modes. Combining ESP and AH is also permitted under the IPsec standards. However, this setup does not significantly improve security [57].

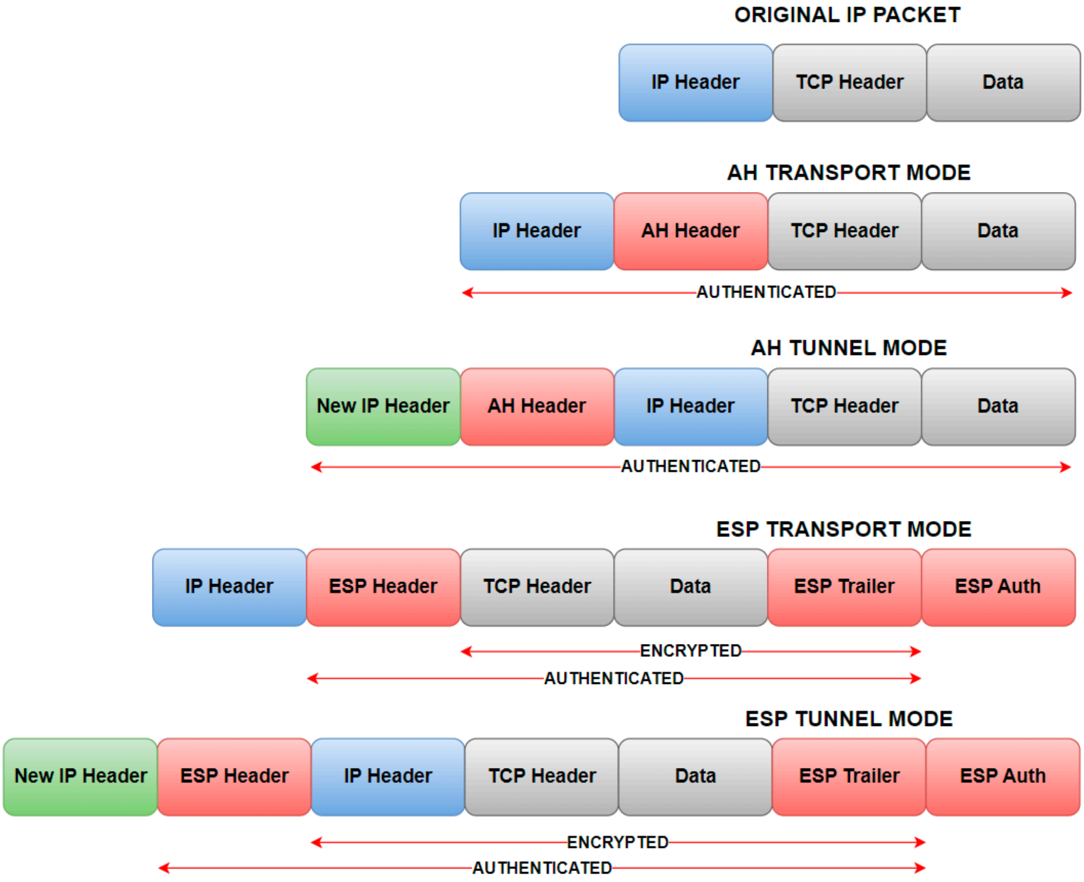


Figure 5. Structure of packets for AH and ESP transport and tunnel modes.

3.2. Proposed Method

For this article, ESP in tunnel mode was explicitly selected for its encryption capabilities, which also extend to the encryption of the IP header.

In ESP tunnel mode, encryption is applied to the original IP packet, preventing any modifications to the encapsulated layers within the ESP. With minimal fields, the ESP header poses challenges for creating a covert storage channel. Figure 6 presents the specific structure of packet fields in this context.

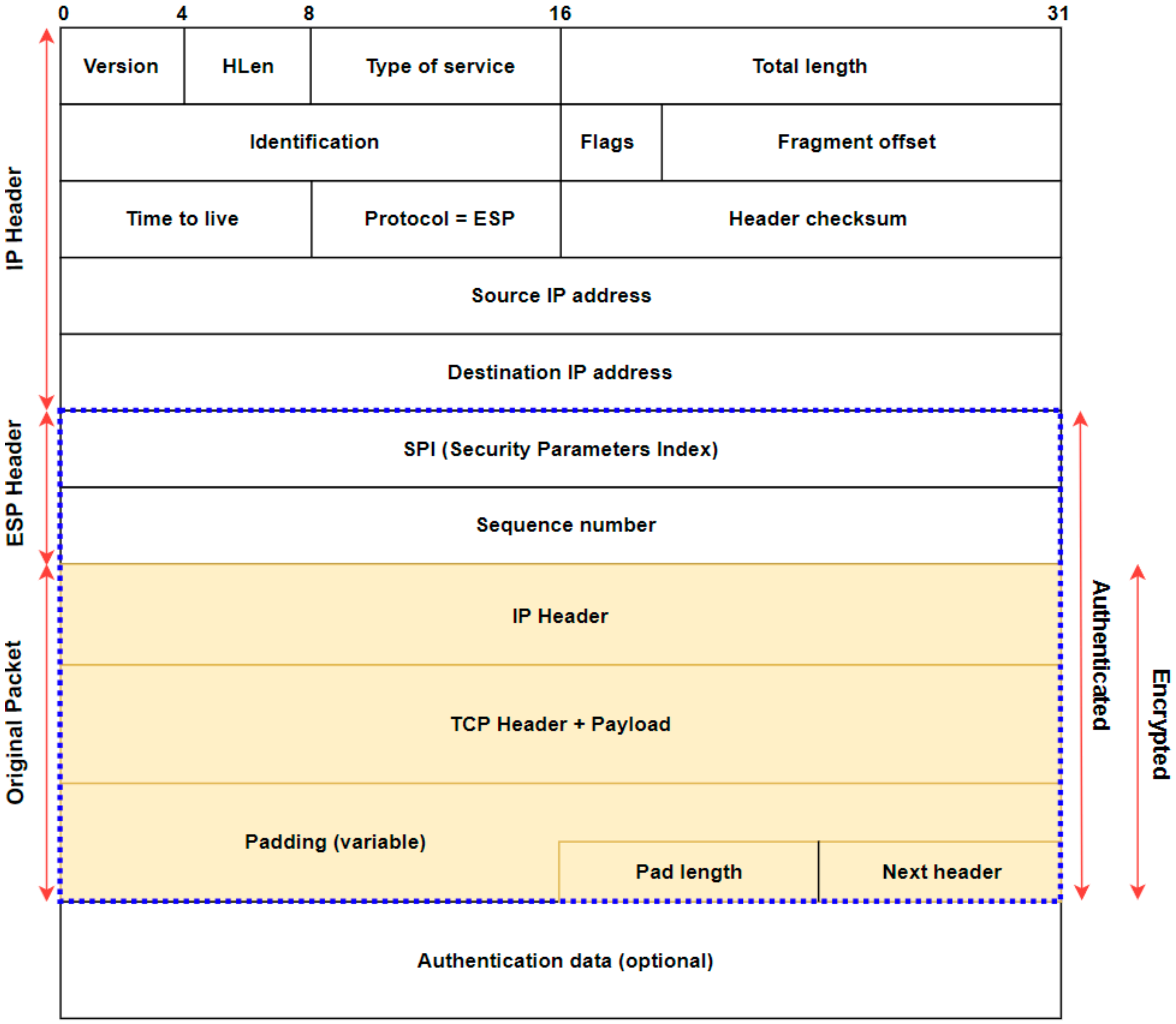


Figure 6. ESP tunnel mode packet structure.

The encryption covers the portion from the IP header to the following header field in the original packet. Any attempts to modify the packet are impossible. Only the end host to which the packet is directed can decrypt it. This results from the encryption algorithm using a unique key to the hosts that have established connections. Furthermore, the Security Parameters Index (SPI) and sequence number in the ESP header are not designed for steganography. SPI is a 32-bit value in the ESP header that uniquely identifies a security association for the communication, the destination IP address, and the security protocol. On the other hand, sequence numbers help prevent replay attacks by ensuring each packet is unique within a specific time frame [58]. Therefore, the SPI must remain unchanged during communication, while the sequence number must be incremented by one for every packet.

The only modifiable fields are those within the newly added IP header, which encapsulates the ESP header. In our approach, we have used the following IPv4 header fields:

1. Identification—this field must be unique for each datagram in a single transmission. This means the field is static only when fragmentation occurs [59]. Therefore, if we

ensure no fragmentation, it is possible to encode 2 bytes of covert data because there is no clear relationship between the generated numbers.

2. Time-to-live—this is an 8-bit field. Its value can range from 0 to 255. However, IPv4 is a best-effort protocol. This, combined with the expansion of the internet, makes it unnecessary to use all 8 bits provided for the TTL field. Therefore, it is possible to use the most significant bits to create a storage covert channel. Its bandwidth depends on whether the network is large or small.

In addition to the identification and time-to-live fields, network packet headers have other essential fields, including flags containing reserved flags and fragment offset, which is important when fragmentation occurs. Another significant field is the type of service (TOS), which is vital in specifying the quality of service for the transmitted data. However, it is important to note that these additional fields have specific purposes and constraints, and their usage could be more flexible.

Considering all possibilities of creating a covert channel within the IPv4 header, we can establish host-host communication where both devices are aware of the steganography communication scheme, ensuring its success. Nonetheless, in most established networks, multiple devices, such as routers, are along the path of packets. The communication schema for everyday traffic communication is illustrated in Figure 7.

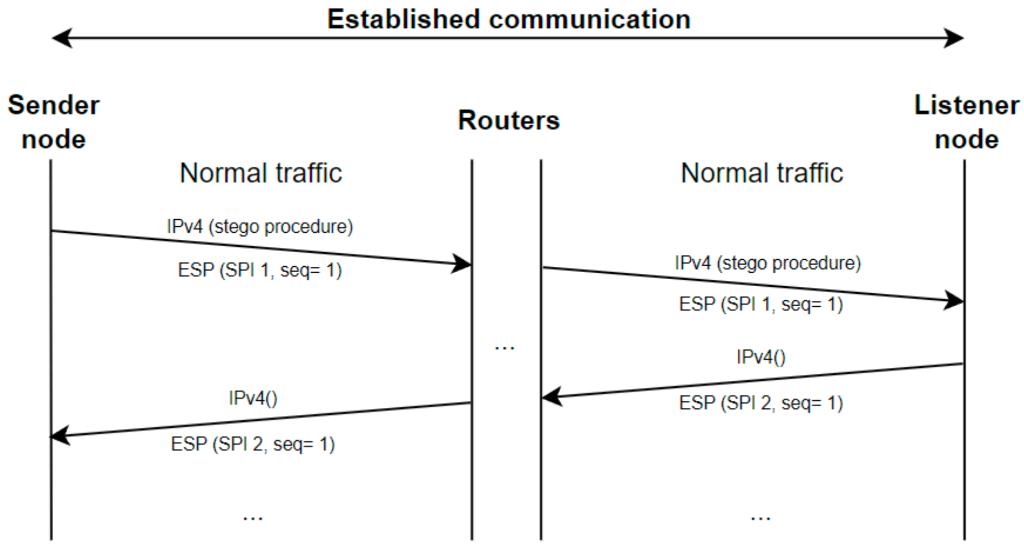


Figure 7. Normal traffic packets sequence.

If any of these devices alters any of the header fields in between, the integrity of the covert channel will be compromised. This challenge can be addressed by implementing the proposed covert channel in the routing device. This implementation can function by either dropping the packet when an incorrect steganography message is received or by transmitting the MAC address of the compromised device to the destination host.

The 48-bit MAC address is covertly embedded within the 16-bit identification field. We distribute the data across three distinct packets to facilitate efficient communication of this information. Moreover, the time-to-live field's most significant bits provide extra information about the identification field. These bits indicate whether the information being transmitted pertains to the first, second, or third segment of the MAC address. If a packet does not contain any part of the MAC address, two zero bits are inserted at the most significant positions of the time-to-live field. This arrangement leads to four possible scenarios:

- The identification field carries the first 16 bits of the MAC address, and the two most significant bits of the time-to-live field are set to 01.
- The identification field carries the second 16 bits of the MAC address, and the two most significant bits of the time-to-live field are set to 10.

- The identification field carries the third 16 bits of the MAC address, and the two most significant bits of the time-to-live field are set to 11.
- The identification field does not include MAC address data, and the two most significant bits of the time-to-live field are set to 00.

After receiving the 48-bit MAC address, the device records it in a table of corrupted MAC addresses. This data is retained until the network controller notifies the hosts that the device with the specified MAC address is no longer considered corrupted.

The packet sequence for sending the MAC address is illustrated in Figure 8.

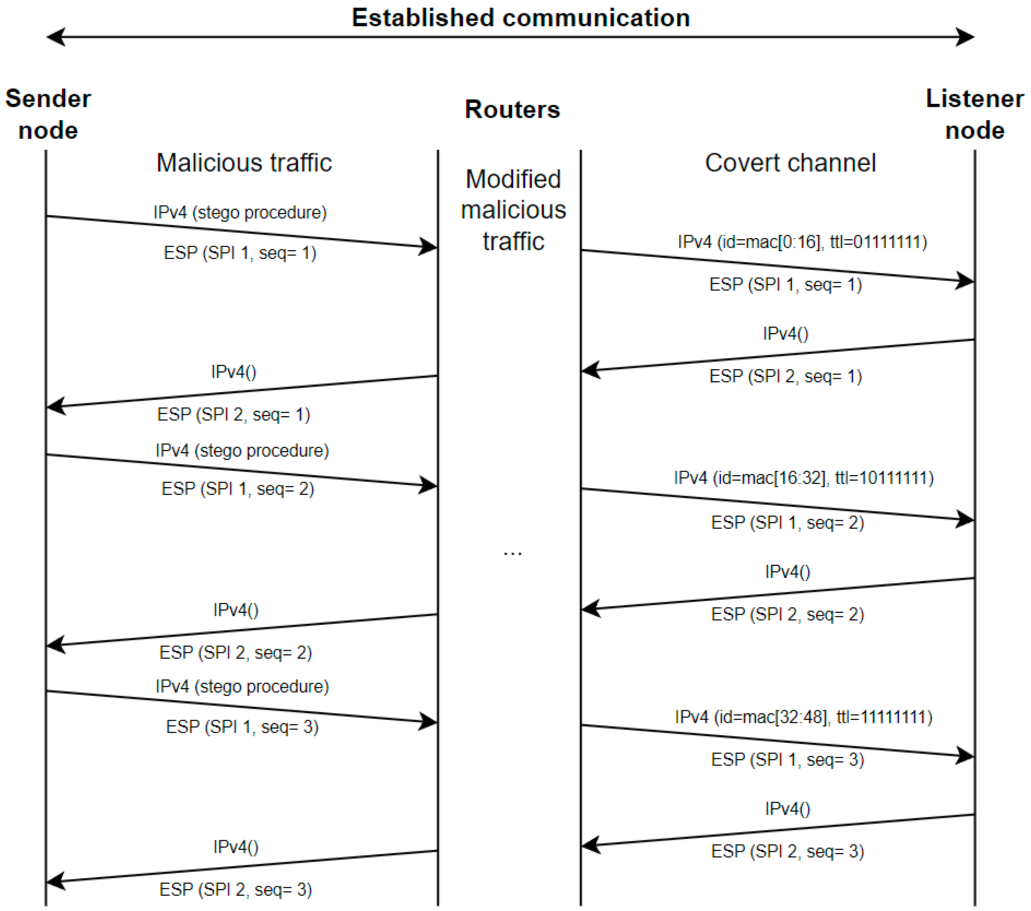


Figure 8. Modified malicious traffic packet sequence.

This MAC address-based approach can alert the network controller about compromised devices. The end host can communicate with the network controller and transmit the MAC address associated with potentially unauthorized access. Alternatively, routers can adopt this mechanism by halting traffic forwarding and sending the MAC address to the controller even before allowing packet forwarding to the end host. As a result, security is enhanced throughout the entire network.

The presented approach bandwidth is equal to a maximum of 3 bytes by utilizing identification and time-to-live fields.

4. Test Scenarios Architecture

IPsec has an integrity system to avoid injecting malicious ESP packets into the SA ESP packets flow: the anti-replay window [60,61].

The principle is to check the packet's sequence information encoded into the ESP header's ESN field. If the security gateway (or host) detects repeated W packets within a reception window, it drops them.

This mechanism preserves the integrity of the inner IP flow after IPsec deencapsulation. Nevertheless, as the control of each packet within a large anti-replay window can require significant CPU consumption of the security gateway and repetition of missing packets and some transmission resources, it is possible to affect the complete decoding of an IPsec flow by flooding the interface of the security gateway with a considerable amount of repeated ESP packets. At least flooding will significantly increase latency and jitter of decoded IPsec application flow.

A standard cyber defense will detect an increase in packets without being able to sort out valid and invalid packets. A security gateway will be able to raise some alerts on an increasing number of dropped packets due to anti-replay without being able to easily identify the hacking machine.

By introducing steganography into ESP flow, we can:

- detect along the IP path, the segment in which the malicious packets are introduced
- segregate before reaching the security, valid, and invalid packets.

Figure 9 describes a classic network architecture in which we interconnect several LANs connected via several routers via IPsec (site-to-site case).

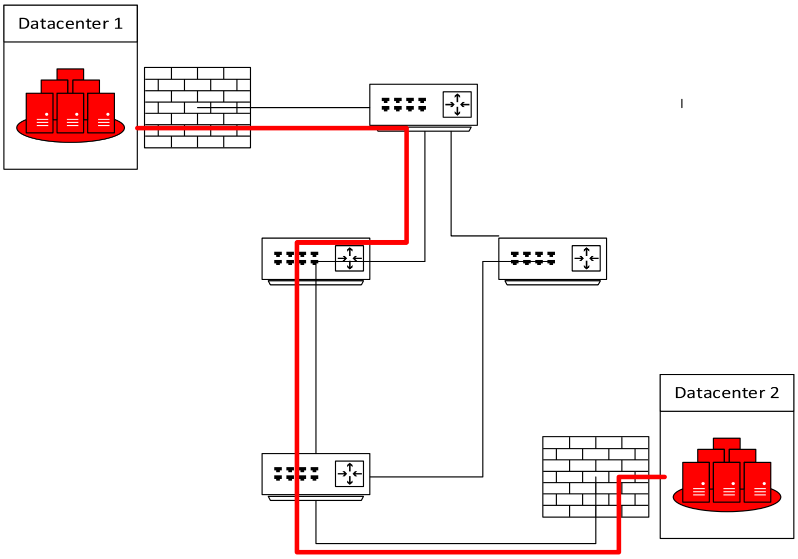


Figure 9. The standard architecture of site-to-site IPsec connection over several networks.

We need at least two probes to control this IPsec flow in one direction. One is at the beginning of the flow, right after the security gateway encodes the steganography message. The other is just before the destination security gateway decodes the steganography message, checks its validity, drops the packet, and alerts the security agent.

This probe deployment option allows for determining whether a DDoS attack is present but does not precisely locate the hacking machine.

Figure 10 gives an example of architecture with this probe deployment option and a hacking machine in the network injecting malicious ESP packets into a router.

To detect the corrupted transmission segment, more probes must be introduced to cover inter-router transmission segments. Additional probes are also needed to cover potential rerouting of IPsec flows through another path going through various routers due to link failure, for example.

Figure 11 shows an example of steganography probe deployment covering all transmission links.

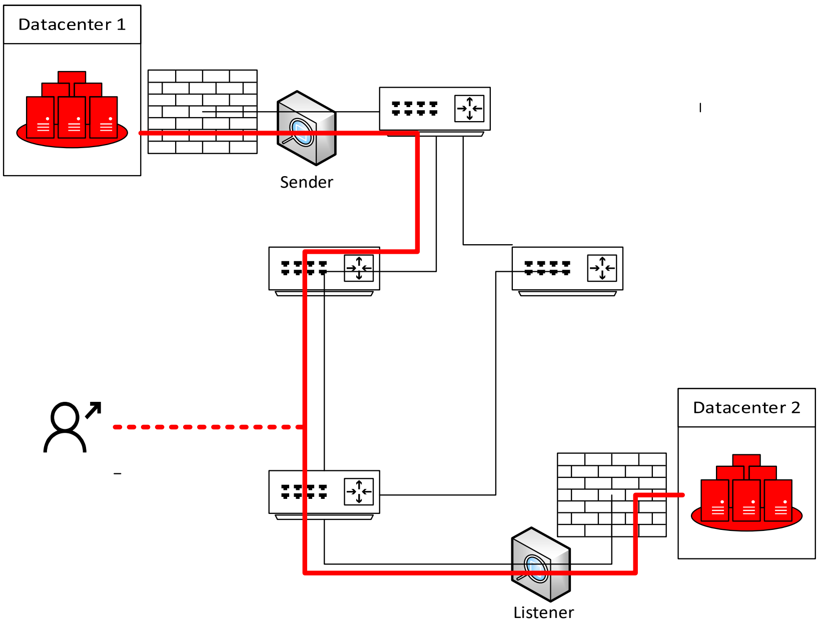


Figure 10. Use case with two probes for controlling an IPsec flow with a hacker.

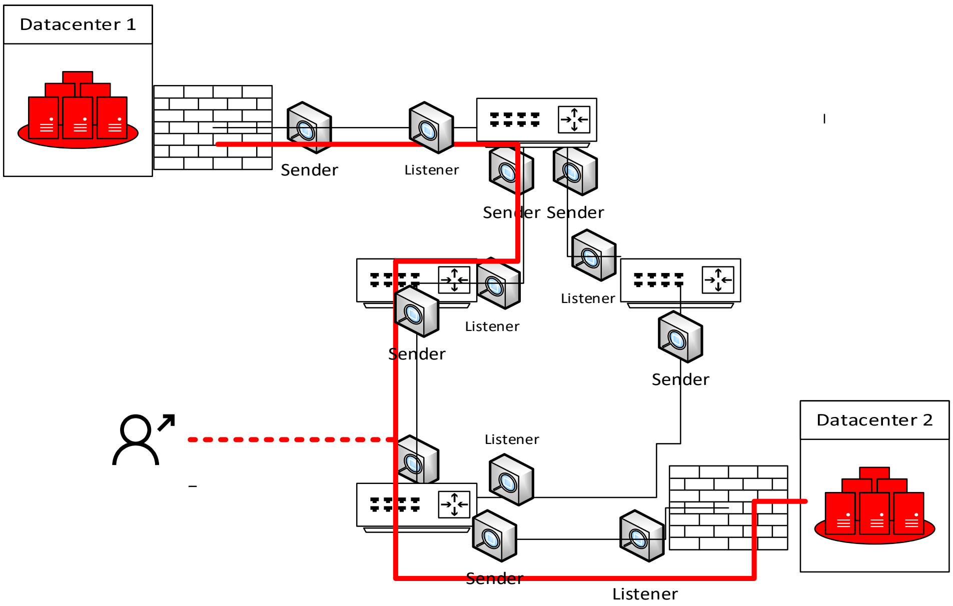


Figure 11. Use case with probes per inter-router link.

This configuration of probes allows for the better detection of malicious packets. For example, the listener probe of the bottom left router will detect that the hacking machine is somewhere on its supervised network segment. It also allows early network protection (either by dropping packets before reaching the last network segment or activating counter-security measures in the last router).

Nevertheless, this deployment scheme of probes introduces additional complexity to message encoding/decoding: who generates the message into the hidden channel created using steganography from a send to a listener probe?

The easiest solution is to manage the message "per proximity": a sender will encode a message for every next listener probe on its next routing table hopes (Figure 12).

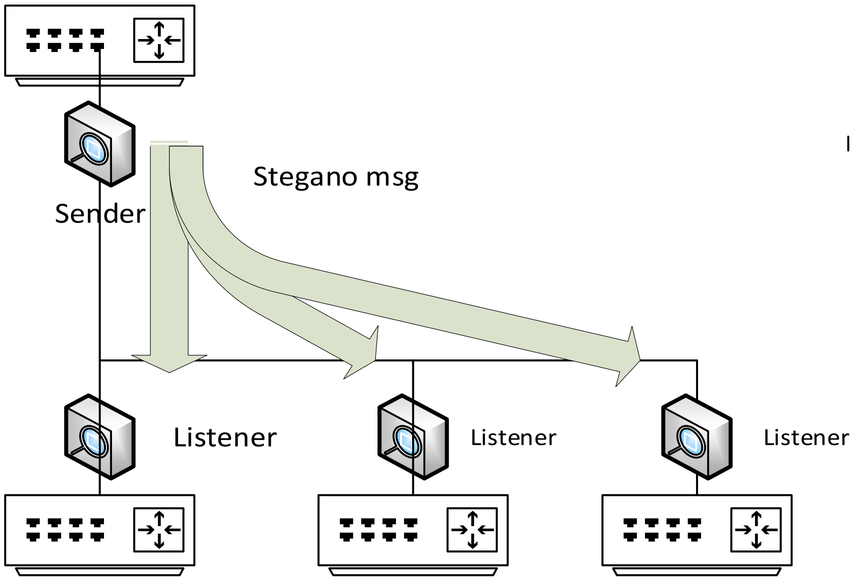


Figure 12. Message encoding/decoding per proximity to the router.

The content of the message sent can be unique for every listener. The next hope is to identify the faulty network segment even more precisely.

A last approach is to have an end-to-end unique message encoding per IPsec flow (from security gateway to destination security gateway) but with probes on each network segment.

In this case, we must introduce a new probe type: the relay probe. The rule of relay probes is to control the received message by knowing the sender and its associated steganography message, drop it or alert the network controller in case of the malicious packet, and eventually regenerate a complete original message if the IP flow has suffered from loss in the previous network segments. The complexity of a relay probe resides in the ability to discriminate and control flows by maintaining check context for associated message decoding. In the previous case, the listener had to check only one message content.

Relay probes can allow checking the corruption of a router. Figure 13 shows an example of deployment with relay probes.

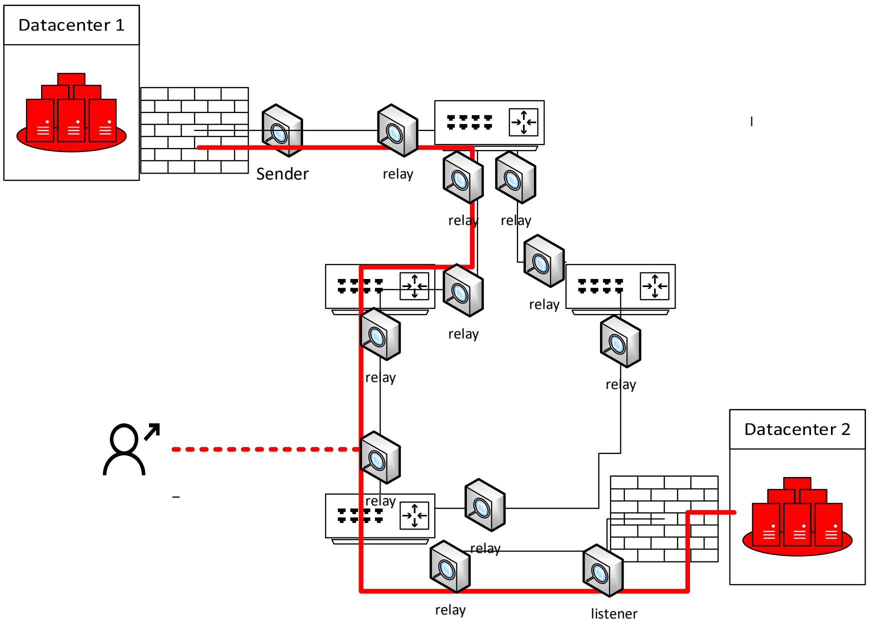


Figure 13. Use case with a mix of sender, listener, and relay probes.

5. Results and Analysis

In most cases where network steganography is employed, maximizing bandwidth, undetectability, and robustness are primary objectives. However, our priorities differ in our approach, which centers on detecting malicious traffic. Rather than emphasizing robustness and undetectability, our focus lies in maximizing bandwidth and the ability to detect changes. We aim to prevent malicious traffic from coincidentally matching predefined signatures. Here, we utilize a 2-byte identification field and allocate 2 bits to the time-to-live field, resulting in an effective bandwidth of 18 bits. If we expand to utilize all bits in the TTL field, the bandwidth increases to 3 bytes, allowing for 2^{24} distinct values. This expansion significantly reduces the chance of overlap with predefined signatures in malicious traffic injection scenarios. Detecting a 3-byte signature that varies with each packet is practically feasible only if the signature generation method is known.

The solution exploits containerization technology via Docker, wherein Docker containers play host to the Visualization and Alert Service components. These components, constructed using Pushgateway, Prometheus, and Grafana, are responsible for visualizing and monitoring metrics to detect malicious traffic. The presented solution funnels metrics obtained during the steganographic message detection process to the Pushgateway. This gateway, serving ephemeral and batch jobs, facilitates Prometheus's metrics collection. Prometheus, acting as a monitoring and alerting toolkit, scrapes and stores metrics, making them queryable and available for alerting. Grafana is employed to craft dashboards that present the collected metrics in a user-friendly format, allowing the creation of customizable visualizations, graphs, and charts.

Within the solution architecture, the Pushgateway is a data ingestion gateway. It facilitates collecting and aggregating time-series metrics generated by both ephemeral and batch jobs. Metrics are "pushed" toward the Pushgateway, which mediates data sources and Prometheus.

Prometheus operates as both a time-series database and a monitoring system. It collects, stores, and querying of metrics from diverse sources, including the Pushgateway. By employing a pull-based model, Prometheus regularly scrapes metrics endpoints, storing the acquired metrics in its data repository for historical analysis and querying.

Grafana complements the solution and serves as a potent visualization and dashboarding platform. It retrieves metric data from Prometheus, empowering users to create personalized, interactive dashboards. Grafana supports various visualization types, such as graphs, charts, and tables, fostering insightful interpretation of monitored metrics.

The synergistic utilization of Pushgateway, Prometheus, and Grafana furnishes a comprehensive solution for real-time monitoring, collection, storage, and visualization of time-series metrics. The network administrator can derive insights into system performance, diagnose issues, and proactively respond to anomalies. Alert capabilities ensure timely notification of critical incidents or deviations from predefined baselines, streamlining efficient incident response and problem resolution. The figures below present the dashboards generated by Grafana.

These dashboards showcase crucial metrics, including the count of packets with detected steganographic content (Metric A) and the count of packets without detected steganographic content (Metric B). These metrics serve as indicators of steganographic message presence and activity. The detection ratio, derived by dividing Metric A by Metric B, offers insights into the efficacy of the steganographic detection process. Grafana facilitates the generation of alarms based on predefined threshold values, triggering notifications or actions when metrics breach specified limits. Elevated values of Metric B and the detection ratio signify potential attacks, establishing it as a valuable tool for monitoring and response (Figure 14).

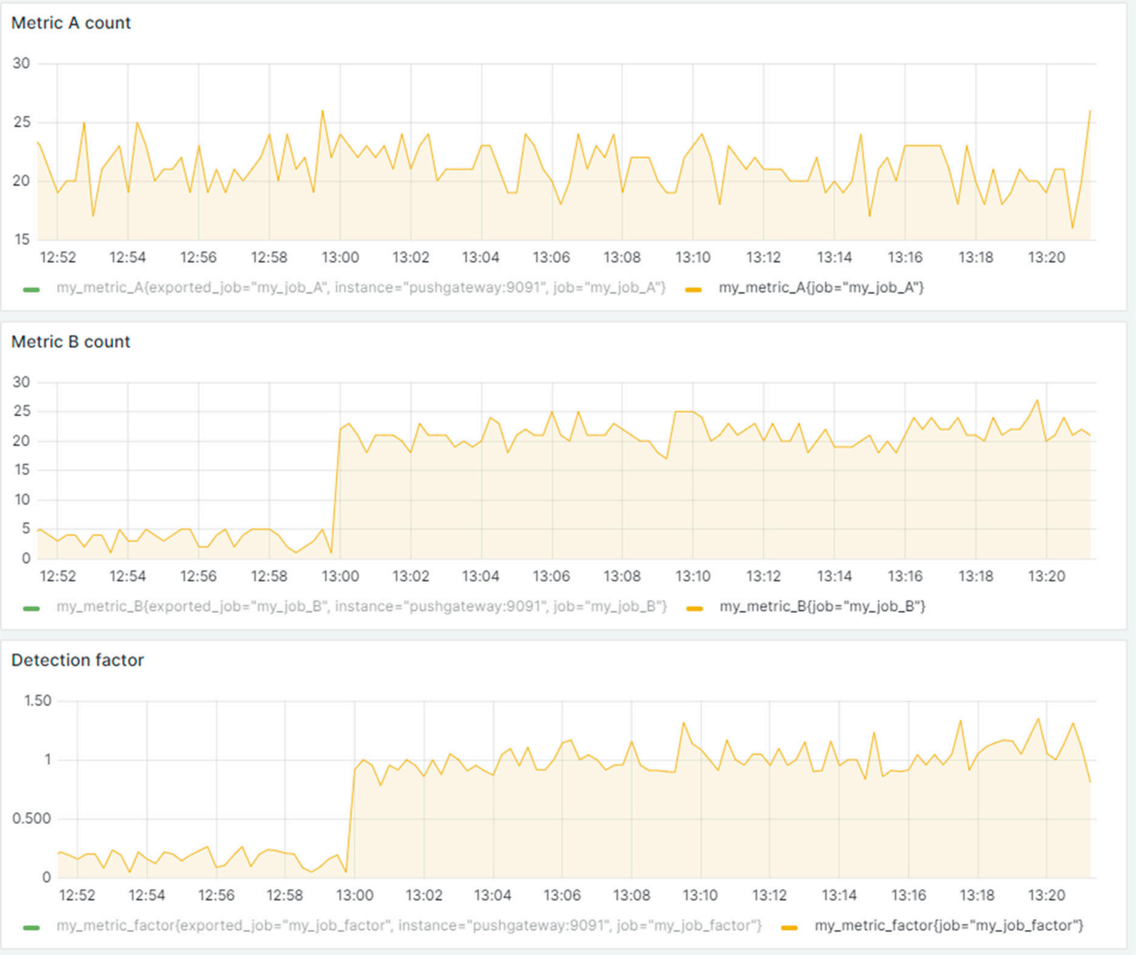
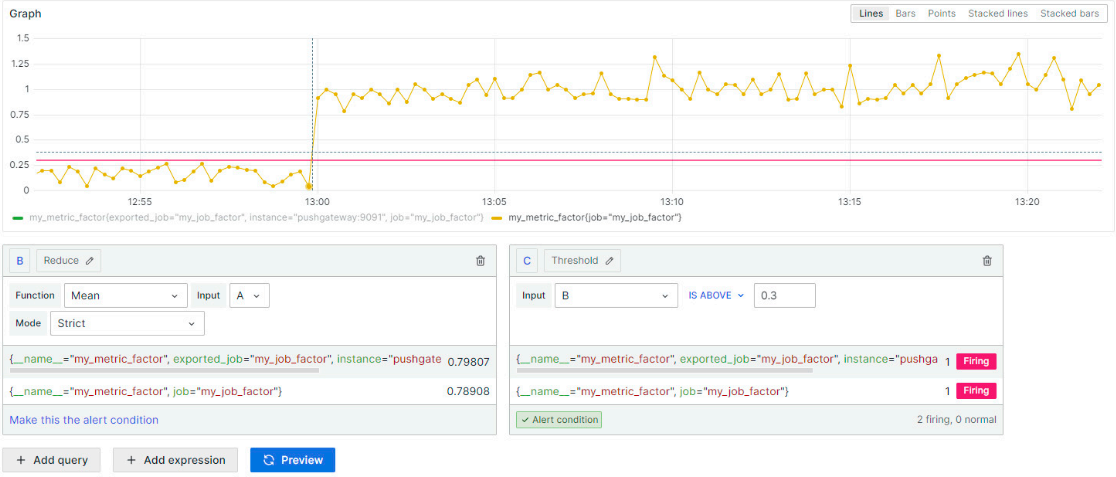


Figure 14. Dashboards in Grafana.

Figure 15 illustrates an exemplary setup of the alarm mechanism, outlining the configuration that dictates the activation criteria for alarms based on metric values. The configuration empowers the establishment of triggers, specifying conditions under which alarms are initiated. These triggers are tied to predefined thresholds and operate within designated time intervals. Such a setup enables administrators to receive alerts promptly in the event of malicious activity.



Function: Mean
Input: A
Mode: Strict

{__name__="my_metric_factor", exported_job="my_job_factor", instance="pushgate 0.79807
{__name__="my_metric_factor", job="my_job_factor"} 0.78908

Make this the alert condition

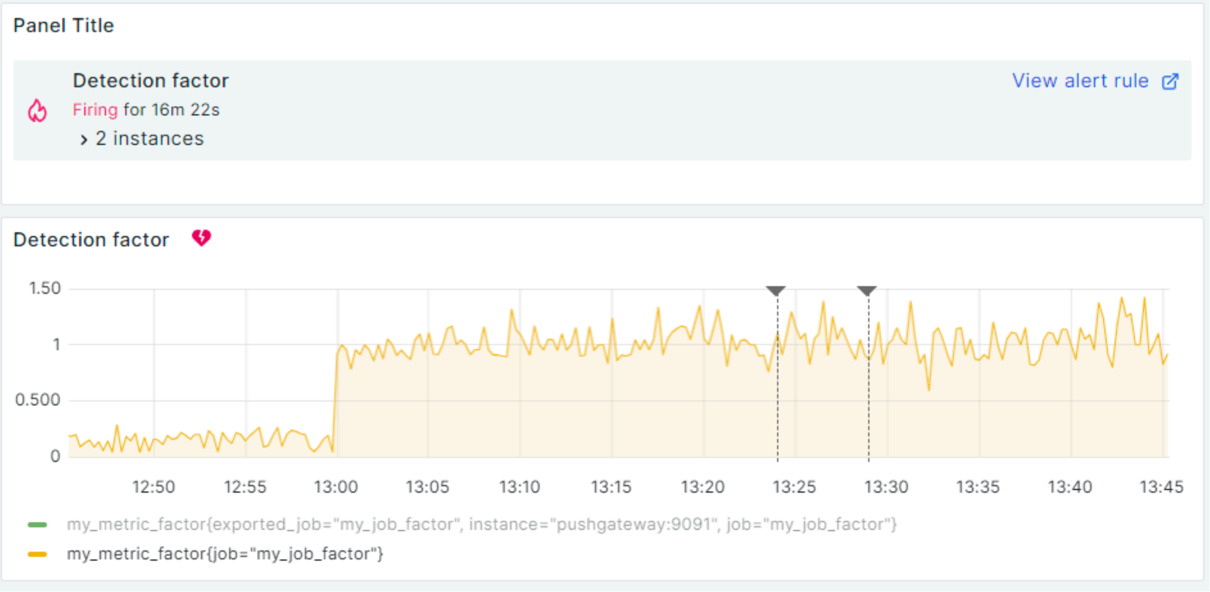
Input: B IS ABOVE 0.3

{__name__="my_metric_factor", exported_job="my_job_factor", instance="pushga 1 Firing
{__name__="my_metric_factor", job="my_job_factor"} 1 Firing

✓ Alert condition 2 firing, 0 normal

Figure 15. Creating threshold alert in Grafana.

The determination of unauthorized traffic relies on the detection threshold. This threshold is predefined and configured by the network administrator, leveraging their expertise and insights into the current conditions within the network. The administrator's knowledge enables fine-tuning the detection mechanism, tailoring it to the unique characteristics of the network, and anticipating potential threats. This approach ensures flexibility in adapting the solution, allowing for a proactive response to evolving network conditions and emerging security challenges. The alarm is triggered once a metric surpasses the set threshold, as depicted in Figure 16.



Panel Title

Detection factor

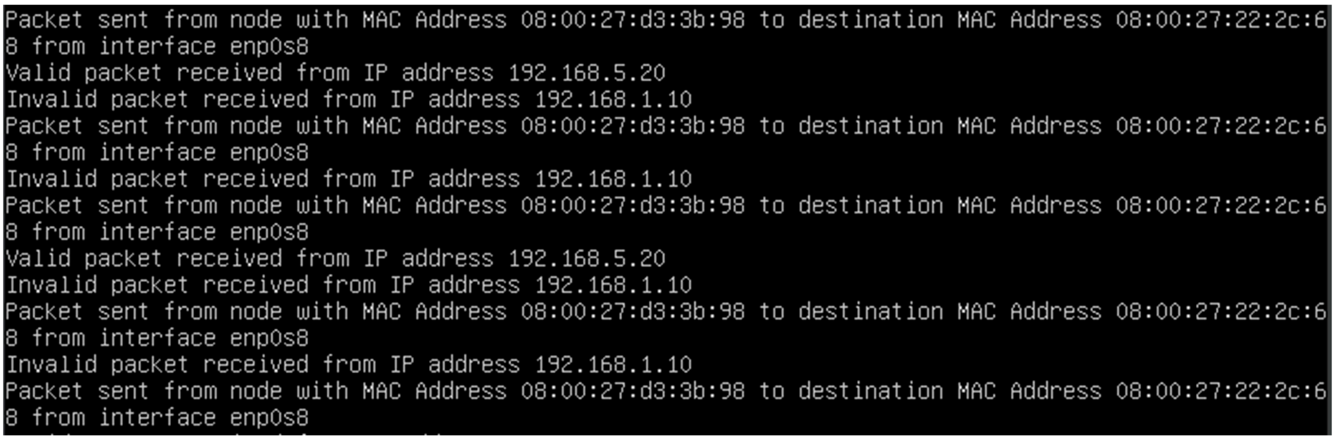
Firing for 16m 22s

› 2 instances

View alert rule
Detection factor
1.50
1
0.500
0
12:50
12:55
13:00
13:05
13:10
13:15
13:20
13:25
13:30
13:35
13:40
13:45
my_metric_factor{exported_job="my_job_factor", instance="pushgateway:9091", job="my_job_factor"}
my_metric_factor{job="my_job_factor"}

Figure 16. Notification for fired alert.

Apart from Grafana, there is also information about the MAC address received by the network element incorporating the described approach. Figure 17 illustrates an example of output information about a malicious node. This output includes the IP address from which an invalid packet was received. Additionally, it provides information about the MAC address of the malicious node and the destination MAC address, along with interface details indicating from which address the packet was sent. This information serves as a basis for various actions within the network. For instance, the network controller can take measures such as excluding the identified host from the network based on this data. An example of MAC address interception output is illustrated in Figure 17.



```
Packet sent from node with MAC Address 08:00:27:d3:3b:98 to destination MAC Address 08:00:27:22:2c:6
8 from interface enp0s8
Valid packet received from IP address 192.168.5.20
Invalid packet received from IP address 192.168.1.10
Packet sent from node with MAC Address 08:00:27:d3:3b:98 to destination MAC Address 08:00:27:22:2c:6
8 from interface enp0s8
Invalid packet received from IP address 192.168.1.10
Packet sent from node with MAC Address 08:00:27:d3:3b:98 to destination MAC Address 08:00:27:22:2c:6
8 from interface enp0s8
Valid packet received from IP address 192.168.5.20
Invalid packet received from IP address 192.168.1.10
Packet sent from node with MAC Address 08:00:27:d3:3b:98 to destination MAC Address 08:00:27:22:2c:6
8 from interface enp0s8
Invalid packet received from IP address 192.168.1.10
Packet sent from node with MAC Address 08:00:27:d3:3b:98 to destination MAC Address 08:00:27:22:2c:6
8 from interface enp0s8
```

Figure 17. For example, MAC address interception output.

In the realm of network security, the presented solution not only identifies but also enhances the network's ability to secure itself. Delving into the specifics, as showcased in the visual representation, the network gains a nuanced understanding of potential threats by capturing detailed information on malicious nodes. This includes key data points such as MAC addresses. When armed with this intelligence, the system can swiftly implement targeted responses, exemplified by dynamically isolating and excluding flagged hosts from the network. Implementing this sophisticated strategy metamorphoses the network into a vigilant protector, reinforcing its resilience against potential security vulnerabilities.

In addition to detection tests, we conducted network overhead tests. The results are depicted in Figure 18, revealing an overhead ranging from 0.3 to a maximum of 1.4 milliseconds. We observed a slight increase in delay time as the volume of packets transmitted over the network increased. This delay is negligible, indicative of our approach introducing minimal overhead. This is attributed to the low computational complexity of our method.

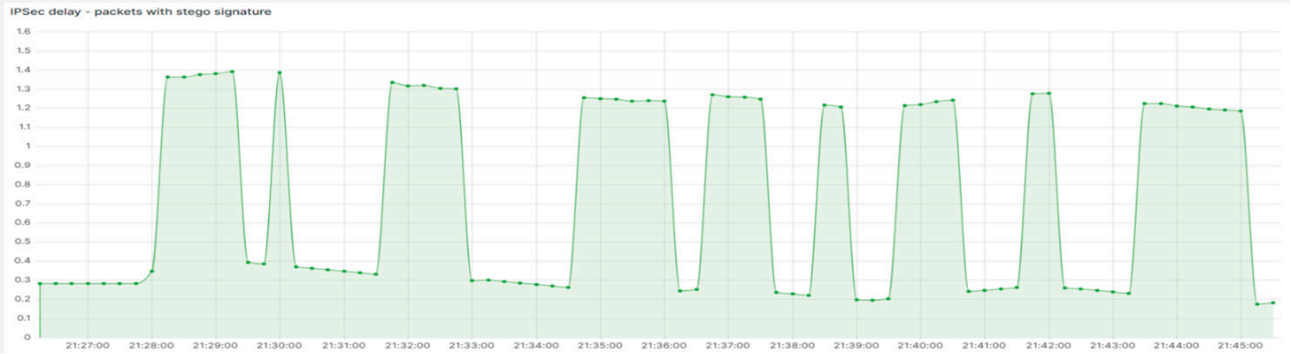


Figure 18. Network overhead test dashboard.

6. Conclusions

In an era characterized by the ever-evolving intricacies of communication protocols, the significance of network steganography is gaining prominence. As communication techniques advance, the need for more sophisticated methods becomes imperative. This paper introduces a network steganography approach, demonstrating that steganography can function as a crucial tool for fortifying the security aspects of data transmission systems. The primary objective of our proposed steganographic technique is to enhance the security of data transmission by enabling the concealment and transmission of specific information, specifically the MAC address. The ultimate goal is to effectively isolate malicious devices and bolster the overall network security infrastructure. With IPSec, we employ a novel steganographic method that leverages the IPv4 protocol fields, precisely the identification and time-to-live parameters. Incorporating these elements into our steganographic technique enables the covert transmission of information within the network. This approach facilitates the targeted exclusion of adversarial devices based on their MAC addresses, strengthening the network's ability to identify and neutralize potential security threats.

In summary, our paper introduces a network steganography method that allows the retrieval of the MAC addresses of compromised devices. By utilizing this technique, we demonstrate the potential for enhancing security measures in data transmission systems by covertly transmitting crucial information and the targeted isolation of malicious devices. This research contributes to the ongoing efforts to fortify network security in an environment marked by the relentless evolution of communication protocols.

Author Contributions: G.J. contributed to IPSec and method description. R.V. contributed to the architecture. D.J. contributed to the Section 5. Z.P., contributed to the Sections 1 and 2 and revision. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was supported and funded by the European Defence Agency, grant no. B 1520 IAP4 GP covering the Ad Hoc Research and technology Project entitled "Software Defined Tactical and Theatre Network".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Author René Veyland was employed by the company Thales. All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Lenarczyk, P.; Piotrowski, Z. Parallel Blind Digital Image Watermarking in Spatial and Frequency Domains. *Telecommun. Syst.* **2013**, *54*, 287–303. [CrossRef]
2. Piotrowski, Z. Drift Correction Modulation Scheme for Digital Signal Processing. *Math. Comput. Model.* **2013**, *57*, 2660–2670. [CrossRef]
3. Jirwan, N.; Singh, A.; Vijay, D.S. Review and Analysis of Cryptography Techniques. *Int. J. Sci. Eng. Res.* **2013**, *4*, 1–6.
4. Shawkat, S.A. *Enhancing Steganography Techniques in Digital Images*; Faculty of Computers and Information, Mansoura University: Mansoura, Egypt, 2016. [CrossRef]
5. Kahn, D. *The Story of Secret Writing*; The Macmillan Company: New York, NY, USA, 1967; Volume 1, p. 10019, ISBN 0-684-83130-9.
6. Kunchur, M.N. The Human Auditory System and Audio. *Appl. Acoust.* **2023**, *211*, 109507. [CrossRef]
7. Kunhoth, J.; Subramanian, N.; Al-Maadeed, S.; Bouridane, A. Video Steganography: Recent Advances and Challenges. *Multimed. Tools Appl.* **2023**, *82*, 41943–41985. [CrossRef]
8. Kaczyński, M.; Piotrowski, Z. High-Quality Video Watermarking Based on Deep Neural Networks and Adjustable Subsquares Properties Algorithm. *Sensors* **2022**, *22*, 5376. [CrossRef] [PubMed]
9. Li, J.; Wang, K.; Jia, X. A Coverless Audio Steganography Based on Generative Adversarial Networks. *Electronics* **2023**, *12*, 1253. [CrossRef]
10. Liu, X.; Li, Z.; Ma, J.; Zhang, W.; Zhang, J.; Ding, Y. Robust Coverless Steganography Using Limited Mapping Images. *J. King Saud. Univ.-Comput. Inf. Sci.* **2022**, *34*, 4472–4482. [CrossRef]
11. Meng, L.; Jiang, X.; Zhang, Z.; Li, Z.; Sun, T. A Robust Coverless Video Steganography Based on Maximum DC Coefficients against Video Attacks. *Multimed. Tools Appl.* **2023**, *83*, 13427–13461. [CrossRef]
12. Lubacz, J.; Mazurczyk, W.; Szczypiorski, K. Principles and Overview of Network Steganography. *IEEE Commun. Mag.* **2014**, *52*, 225–229. [CrossRef]
13. Grzesiak, K.; Piotrowski, Z.; Kelner, J.M. A Wireless Covert Channel Based on Dirty Constellation with Phase Drift. *Electronics* **2021**, *10*, 647. [CrossRef]
14. Grzesiak, K.; Piotrowski, Z.; Kelner, J.M. Covert Channel Based on Quasi-Orthogonal Coding. *Electronics* **2023**, *12*, 2249. [CrossRef]
15. Piotrowski, Z. Angle Phase Drift Correction Method Effectiveness. In Proceedings of the Signal Processing Algorithms, Architectures, Arrangements, and Applications SPA 2009, Poznan, Poland, 24–26 September 2009; pp. 82–86.
16. Piotrowski, Z.; Wojtuń, J.; Kamiński, K. Subscriber Authentication Using GMM and TMS320C6713DSP. *Prz. Elektrotechniczny* **2012**, *88*, 127–130.
17. Szczypiorski, K. HICCUPS: Hidden Communication System for Corrupted Networks. *Int. Multi-Conf. Adv. Comput. Syst.* **2003**, 31–40. Available online: https://www.gray-world.net.brrll.com/papers/acs2003-hiccups.pdf (accessed on 25 March 2024).
18. Simmons, G.J. The Prisoners' Problem and the Subliminal Channel. In *Advances in Cryptology: Proceedings of Crypto 83*; Chaum, D., Ed.; Springer US: Boston, MA, USA, 1984; pp. 51–67, ISBN 978-1-4684-4730-9.
19. Backer, C. *Subliminal Channels in Cryptographic Systems*; Seminararbeit; Ruhr-Universität Bochum: Bochum, Germany, 2009.
20. Craver, S. On Public-Key Steganography in the Presence of an Active Warden. In *Information Hiding*; Aucsmith, D., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1525, pp. 355–368, ISBN 978-3-540-65386-8.
21. Caviglione, L. Trends and Challenges in Network Covert Channels Countermeasures. *Appl. Sci.* **2021**, *11*, 1641. [CrossRef]
22. US Department of Defense Department of Defense Trusted Computer System Evaluation Criteria. In *The 'Orange Book' Series*; US Department of Defense, Ed.; Palgrave Macmillan UK: London, UK, 1985; pp. 1–129, ISBN 978-0-333-53947-7.
23. Celina, B. Covert Channels in Computer Networks. *INSA TC* 2020. Available online: https://medium.com/insa-tc/covert-channels-in-computer-networks-26a33fd911b2 (accessed on 25 March 2024).
24. Sawicki, K.; Bieszczad, G.; Piotrowski, Z. StegoFrameOrder—MAC Layer Covert Network Channel for Wireless IEEE 802.11 Networks. *Sensors* **2021**, *21*, 6268. [CrossRef] [PubMed]
25. Uttarakhand Technical University; Bora, G.; Bora, S.; Singh, S.; Arsalan, S.M. OSI Reference Model: An Overview. *IJCTT* **2014**, *7*, 214–218. [CrossRef]

26. Szczypiorski, K.; Mazurczyk, W. Steganography in IEEE 802.11 OFDM Symbols. *Secur. Commun. Netw.* **2016**, *9*, 118–129. [CrossRef]
27. Martins, D.; Guyennet, H. Attacks with Steganography in PHY and MAC Layers of 802.15.4 Protocol. In Proceedings of the 2010 Fifth International Conference on Systems and Networks Communications, Nice, France, 22–27 August 2010; pp. 31–36.
28. Yamaguchi, R.; Ochiai, H.; Shikata, J. A Physical-Layer Security Based on Wireless Steganography Through OFDM and DFT-Precoded OFDM Signals. In Proceedings of the 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 25–28 May 2020; pp. 1–5.
29. Bedi, P.; Dua, A. Network Steganography Using the Overflow Field of Timestamp Option in an IPv4 Packet. *Procedia Comput. Sci.* **2020**, *171*, 1810–1818. [CrossRef]
30. Kundur, D.; Ahsan, K. Practical Internet Steganography: Data Hiding in IP. *Proc. Tex. Wksp. Secur. Inf. Syst.* **2003**. Available online: https://ww2.comm.utoronto.ca/~dkundur//pub_pdfs/KunAhsTXSecWrkshp03.pdf (accessed on 25 March 2024).
31. Kheddar, H.; Bouzid, M. Implementation of Covert Channel Method Based on IPv4 Identification Field over NS-3. *Int. J. Eng. Res. Appl.* **2015**, *5*, 44–48.
32. Mazurczyk, W.; Szczypiorski, K. Steganography in Handling Oversized IP Packets. In Proceedings of the 2009 International Conference on Multimedia Information Networking and Security, Wuhan, China, 18–20 November 2009; pp. 559–564.
33. Murdoch, S.J.; Lewis, S. Embedding Covert Channels into TCP/IP. In *Information Hiding*; Barni, M., Herrera-Joancomartí, J., Katzenbeisser, S., Pérez-González, F., Eds.; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin/Heidelberg, Germany, 2005; Volume 3727, pp. 247–261, ISBN 978-3-540-29039-1.
34. Miller, B. Steganography in IPV6. Computer Science and Computer Engineering Undergraduate Honors Theses. Available online: https://scholarworks.uark.edu/csceuht/22 (accessed on 25 March 2024).
35. Bobade, S.; Goudar, R.M. Survey and Design Approach of Protocol Steganography in IPv6. *IJCA* **2013**, *69*, 31–34. [CrossRef]
36. Salih, A. *An Adaptive Approach to Detecting Behavioural Covert Channels in IPv6*; School of Science and Technology, Nottingham Trent University: Nottingham, UK, 2017.
37. Pshanoop/ICMPStegano: Network Steganography Tool for ICMP Protocol. Available online: https://github.com/pshanoop/ICMPStegano (accessed on 22 March 2024).
38. ARPNetSteg: Network Steganography Using Address Resolution Protocol. *Int. J. Electron. Telecommun.* **2023**. [CrossRef]
39. Jankowski, B.; Mazurczyk, W.; Szczypiorski, K. PadSteg: Introducing Inter-Protocol Steganography. *Telecommun. Syst.* **2011**, *52*, 1101–1111. [CrossRef]
40. Mazurczyk, W.; Smolarczyk, M.; Szczypiorski, K. Retransmission Steganography Applied. In Proceedings of the 2010 International Conference on Multimedia Information Networking and Security, Nanjing, China, 4–6 November 2010; pp. 846–850.
41. Dhobale, D.D.; Ghorpade, V.R.; Patil, B.S.; Patil, S.B. Steganography by Hiding Data in TCP/IP Headers. In Proceedings of the 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), Chengdu, China, 20–22 August 2010; pp. V4-61–V4-65.
42. Handel, T.G.; Sandford, M.T. Hiding Data in the OSI Network Model. In *Information Hiding*; Anderson, R., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1174, pp. 23–38, ISBN 978-3-540-61996-3.
43. Nair, A.S.; Kumar, A.; Sur, A.; Nandi, S. Length Based Network Steganography Using UDP Protocol. In Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks, Xi'an, China, 27–29 May 2011; pp. 726–730.
44. Fraczek, W.; Mazurczyk, W.; Szczypiorski, K. Stream Control Transmission Protocol Steganography. In Proceedings of the 2010 International Conference on Multimedia Information Networking and Security, Nanjing, China, 4–6 November 2010; pp. 829–834.
45. Mazurczyk, W.; Szczypiorski, K. Covert Channels in SIP for VoIP Signalling. In *Global E-Security*; Jahankhani, H., Revett, K., Palmer-Brown, D., Eds.; Communications in Computer and Information Science; Springer: Berlin/Heidelberg, Germany, 2008; Volume 12, pp. 65–72, ISBN 978-3-540-69402-1.
46. Mehic, M.; Slachta, J.; Voznak, M. Hiding Data in SIP Session. In Proceedings of the 2015 38th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 9–11 July 2015; pp. 1–5.
47. Bender, W.; Gruhl, D.; Morimoto, N.; Lu, A. Techniques for Data Hiding. *IBM Syst. J.* **1996**, *35*, 313–336. [CrossRef]
48. Mazurczyk, W. Lost Audio Packets Steganography: The First Practical Evaluation. *Secur. Commun. Netw.* **2012**, *5*, 1394–1403. [CrossRef]
49. Mazurczyk, W.; Szaga, P.; Szczypiorski, K. Using Transcoding for Hidden Communication in IP Telephony. *Multimed. Tools Appl.* **2014**, *70*, 2139–2165. [CrossRef]
50. Dimitrova, B.; Mileva, A. Steganography of Hypertext Transfer Protocol Version 2 (HTTP/2). *JCC J. Comput. Commun.* **2017**, *5*, 98–111. [CrossRef]
51. Lucena, N.B.; Pease, J.; Yadollahpour, P.; Chapin, S.J. Syntax and Semantics-Preserving Application-Layer Protocol Steganography. In *Information Hiding*; Fridrich, J., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germay, 2004; Volume 3200, pp. 164–179, ISBN 978-3-540-24207-9.
52. Sawicki, K.; Piotrowski, Z. The Proposal of IEEE 802.11 Network Access Point Authentication Mechanism Using a Covert Channel. In Proceedings of the 2012 19th International Conference on Microwaves, Radar & Wireless Communications, Warsaw, Poland, 21–23 May 2012; pp. 656–659.
53. Mazurczyk, W.; Wendzel, S.; Azagra Villares, I.; Szczypiorski, K. On Importance of Steganographic Cost for Network Steganography: On Importance of Steganographic Cost for Network Steganography. *Secur. Comm. Netw.* **2016**, *9*, 781–790. [CrossRef]

54. Kundu, A. Mitigation of Storage Covert Channels in IPSec for QoS Aware Applications. *Procedia Comput. Sci.* **2015**, *54*, 102–107. [CrossRef]
55. On Web-Security and -Insecurity: Practical Bleichenbacher Attacks on IPsec IKE. Available online: https://web-in-security.blogspot.com/2018/08/practical-bleichenbacher-attacks-on-ipsec-ike.html (accessed on 22 March 2024).
56. RFC 6071—IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. Available online: https://datatracker.ietf.org/doc/html/rfc6071 (accessed on 22 March 2024).
57. Authentication Header (AH) and Encapsulating Security Payload (ESP) in IPsec VPNs. Available online: https://help.stonesoft.com/onlinehelp/StoneGate/SMC/6.7.0/GUID-8B599235-C75B-4C67-81FD-F4512FD4806A.html (accessed on 22 March 2024).
58. Encapsulated Security Payload (ESP)—The IMS: IP Multimedia Concepts and Services, Second Edition [Book]. Available online: https://www.oreilly.com/library/view/the-ims-ip/9780470019061/9780470019061_encapsulated_security_payload_open_paren.html (accessed on 22 March 2024).
59. RFC 791—Internet Protocol. Available online: https://datatracker.ietf.org/doc/html/rfc791 (accessed on 22 March 2024).
60. RFC 4302—IP Authentication Header. Available online: https://datatracker.ietf.org/doc/html/rfc4302 (accessed on 19 December 2023).
61. RFC 4303—IP Encapsulating Security Payload (ESP). Available online: https://datatracker.ietf.org/doc/html/rfc4303 (accessed on 22 March 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.