Eliminating Steganography in Internet Traffic with Active Wardens

Gina Fisk[†§], Mike Fisk[†], Christos Papadopoulos[§], and Joshua Neil[†§]

[†]Los Alamos National Laboratory [§]University of Southern California

Abstract. Active wardens have been an area of postulation in the community for nearly two decades, but to date there have been no published implementations that can be used to stop steganography as it transits networks. In this paper we examine the techniques and challenges of a high-bandwidth, unattended, real-time, active warden in the context of a network firewall. In particular, we concentrate on *structured carriers* with objectively defined semantics, such as the TCP/IP protocol suite rather than on the subjective, or *unstructured* carriers such as images that dominate the information hiding literature. We introduce the concept of *Minimal Requisite Fidelity* (MRF) as a measure of the degree of signal fidelity that is both acceptable to end users and destructive to covert communications. For *unstructured carriers*, which lack objective semantics, wardens can use techniques such as adding noise to block subliminal information. However, these techniques can break the overt communications of *structured carriers* which have strict semantics. We therefore use a specification-based approach to determine MRF. We use MRF to reason about opportunities for embedding covert or subliminal information in network protocols and develop both software to exploit these channels, as well as an active warden implementation that stops them. For unstructured carriers, MRF is limited by human perception, but for structured carriers, well known semantics give us high assurance that a warden can completely eliminate certain subliminal or covert channels.

1 Introduction

Network security is one of the most pressing and difficult problems facing modern private organizations and governments. In addition to the daily barrage of unwanted traffic from network scans, viruses, worms, exploit tools, and other unauthorized attempts to gain access, sites must be concerned with malicious insiders using digital carriers to secretly disperse information through the very perimeter that is supposed to be protecting the network. The ubiquitous use of protocols and file structures laden with loose semantics and unused or marginally-significant bits that can be freely used for covert communication channels only furthers those challenges.

This paper focuses on the pragmatic challenges of implementing an active warden as a part of a network firewall. In particular, we concentrate on *structured carriers* such as the TCP/IP protocol suite rather than on the subjective, or *unstructured* carriers, such as images, that dominate the information hiding literature. We call a carrier *structured* if there is a well-defined, objective semantics defining the overt information content of the carrier.

Although wardens have been an area of research in the community since Simmons's 1983 paper [31], after nearly two decades, active wardens still remain largely theoretical. The first contribution of this paper is the creation of an active warden system that operates on network traffic like a firewall. The context of a high-bandwidth, real-time, unattended firewall constrains the approaches available to warden design. For instance, passive detection systems such as intrusion detection systems are often viewed as a second-tier defense, with preventative firewall systems being the preferred primary defense. Further, the practical capabilities of an active warden differ significantly from those of a theoretical warden. The implementation of active wardens will improve understanding of what wardens can do, as well as enabling validation of watermark and steganographic algorithm robustness.

Internet traffic also affects the relative importance of various carriers for covert or subliminal information. While the information hiding community is attentive to the detectability and robustness of information embedded in carriers such as images, audio, and natural language, there has been less effort placed on understanding the ability to use network protocols as carriers. A network warden must address all of the indispensable media types, and clearly protocols themselves cannot be avoided.

Thus, our second contribution is the identification and exploration of a class of carriers that differ significantly from common, *unstructured carriers* exemplified by images, audio, and natural language. Network protocols and computer languages (such as XML, machine code, etc.) are examples of *structured carriers* that are interpreted by machines rather than humans. In contrast, unstructured carriers are subjectively interpreted by a human. While pseudo-random noise has often been suggested as a way to remove subliminal information from a carrier, random noise can also obliterate the overt function of structured carriers. It is for these same reasons that lossy compression is primarily used only on unstructured carriers. As shown later, a more delicate and directed approach is therefore required.

Our third contribution is the concept of *Minimal Requisite Fidelity (MRF)*, which we define as the degree of signal fidelity that is both acceptable to end users and destructive to covert communications. MRF determines the limit of distortion we can introduce to a carrier channel in an attempt to foil any covert or subliminal channels. For the defender, MRF defines an upper-bound in the amount of modifications to the channel. This gives an advantage to the defender, because potentially this can greatly decrease or eliminate the capacity of the covert channel. For unstructured carriers that lack well-defined syntax or semantics, MRF is defined by human perception, but for structured carriers, well-defined semantics give us high assurance that a warden can completely eliminate certain subliminal or covert channels. Further, the MRF paradigm, as applied to network packets, includes an emerging area of research into exploiting and correcting ambiguities that create opportunities for intrusion detection evasion [24, 29].

Our fourth contribution is a specification-based analysis of TCP and IP in order to identify ambiguities that allow network traffic to be used as a carrier for steganographic content. We demonstrate these ambiguities with a software tool that sends covert information in several of these fields. In addition, we implement a network warden that removes these ambiguities without breaking overt communications. We show that TCP and IP are fertile ground for covert and subliminal information, but we also validate

that many of these opportunities can be eliminated through the use of a fully-automated, real-time, network warden. Further, the modifications that a warden must make on packets are generally no more intrusive than those made by existing packet scrubbers [21, 10], firewalls, and NAT boxes.

This paper is organized as follows: In §2, we discuss the threat model and potential consequences. In §3, we summarize the research to date and related work in the relevant areas of steganography and network intrusions. In §4, we explore the concept of *Minimal Requisite Fidelity* and in §5 we introduce examples of and embedding techniques for *unstructured carriers* and *structured carriers*. In §6, we develop algorithmic techniques for enforcing Minimal Requisite Fidelity in IP and TCP, and also examine applicability to other network security problems. We discuss our implementation and challenges that we discovered during its creation in §7, and lastly, we conclude the paper in §8.

2 Threat Model

Historically, the malicious insider has been one of the greatest threats to organizations [2], but techniques to stop these insiders are often time-consuming and inadequate. In the example of the admitted FBI spy Robert Hanssen, his espionage activities were not detected and stopped for over a decade [11, 27]. Meanwhile, he distributed some of the US government's most classified information directly into the hands of the KGB. According to his own affidavit, Hanssen's success was facilitated by various forms of steganography and other undercover techniques to communicate and transfer information.

Secure organizations go to great lengths to secure their machines and networks from outside attackers. However, the vast number of insiders are largely trusted in order to maintain productivity. As a result, most insiders are able to gain complete control of several internal computer systems. Inevitably, there is some communication between these systems and external systems that may cooperate in the transfer of covert data. Since the insiders have access to both restricted data and machines which they can use to covertly distribute that data, the problem of detecting and stopping this unwanted behavior is extremely challenging.

Even where personnel security is not of great concern, malicious software agents provide equivalent threats. There are many paths for viruses, worms, etc. to enter a network. Once active, these agents have all the electronic capabilities of a malicious individual. Further, network communications may be the only communications path these agents have.

Our model is designed for high-security environments where the network is not a free channel, but is instead frequently monitored or restricted against unauthorized usage. Wardens are not a form of censorship themselves, but merely enforce that all communications are overt. We recognize that our framework may not be appropriate for the Internet as a whole, but only for restricted environments where there is a definite threat that a malicious insider could do permanent damage.

In addition to using covert channels in Internet traffic, there are a plethora of other ways that a malicious insider could extract data from a given site, such as copying

Protocol	Covert Channel	Bandwidth	Rule	State?
IP	Hide data in padding bits	31 bits/packet	Zero these bits	No
IP	Use IP id as covert channel	16 bits/packet	Reassemble and randomize IP id	Yes
IP	Set false source address to bounce messages	32 bits/packet	Egress filtering	No
IP	Use of IP timestamp option	1 bit/packet	Require or prohibit usage	No
IP	Use destination address as flag	8 bits/packet	Anomaly Detection	Yes
IP	Use unnecessary fields (ToS, options, DF if a fragment, etc)	varies	Zero these fields	No
TCP	Hide data in padding bits	31 bits/packet	Zero all padding bits	No
TCP	Use initial sequence number as covert channel	32 bits/connection	Apply offset to sequence numbers	No
TCP	Use urgent pointer field when URG bit not set	16 bits/packet	Zero urgent pointer when URG bit is not set	No
TCP	Use reserved bits	6 bits/packet	Zero reserved bits	No
TCP	Use data field when RST=1	varies	Zero data if RST=1	No
TCP	Use source and destination ports as data	32 bits/packet	Anomaly Detection	Yes
UDP	Use checksum use as a flag	1 bit/packet	Anomaly Detection	Yes
ICMP	Use code field when just sending a type	8 bits/packet	Zero code field if just sending a type	No
ICMP	Use unused bits as a covert channel	32 bits/packet	Zero unused bits	No

Fig. 1. Partial rule sets for IP, TCP, UDP, and ICMP

the data onto a floppy disk and carrying it home. However, our research is limited to network security and not physical security, and as such, we do not address this threat.

2.1 Potential Damage

Each steganographic algorithm has a data to carrier ratio that defines the bandwidth of the carrier. For images, this can be very high (over 50%). Using specific embedding algorithms such as BPCS [19], a 1 MB image could contain up to 500K of hidden data.

Perhaps the highest-bandwidth carrier is network traffic itself. Each packet has the potential to carry over 8 bytes of data (See Figure 1). We analyzed traffic at one large site where over 500 million packets left the site each day. Assuming a malicious insider could control timing of packets to get 1 bit of data out per packet, the site could lose over 26 GB annually. If a malicious insider could manipulate 8 bytes in each packet, the site could loose over 4 GB daily. When combined with the potential that each e-mail attachment can have embedded data, the potential loss rates are staggering.

Completely eliminating the leakage of information is very difficult. However, we are primarily concerned with relatively high-bandwidth leakage of information. While it is feasible that a malicious insider could transit a very small amount of information per day that would not be stopped by our warden, we feel that this is a problem that is best addressed once the high-bandwidth leakage is curbed.

2.2 Deficiencies of Detection Techniques

We argue that security mechanisms must adopt *proactive* as well as *reactive* defenses. As evidenced by the need for both firewalls and intrusion detection, prevention is an equally, if not more important problem than detection. While there are several techniques currently in use that reactively attempt to detect steganography in images, this is understandably an impossible task to complete, as there are many places to hide data, and many ways to hide the data in those places. In this section we discuss some of these detection techniques and their limitations.

Johnson has done extensive work in identifying signatures for specific steganographic techniques [12, 15]. By closely monitoring the artifacts left from several commercial products, he noticed several distinguishing traits for many commercial products. Since all of the commercial techniques modify the carrier in some way, he was able to document many of these signatures. However, his published observations were limited to commercial steganographic packages, and the majority of malicious insiders would not opt to use such public techniques in their covert transfers.

In [8], Fridrich et al. discuss a simple technique to detect hidden information in the least-significant bits of images by observing the number *close colors* in images. While their technique works relatively well to detect large hidden messages, small embedded messages produce an error rate of up to 40%.

In [28], Provos attempted to find images containing steganography on the Internet. He downloaded 3 million JPEG images from Ebay and Usenet, and performed several tests to attempt to determine if they had embedded data from *JP-Hide* [16], *J-Steg* [17], or *Outguess* [23]. Provos' detection tool *Stegdetect* [33] identified over 54,000 images with these detection signatures, but was unable to find the passwords for any of these images. As such, his results were inconclusive.

While it is useful to gain intelligence about the activities of attackers, our primary goal is to provide system and information security rather than to collect attack information or spend time finding and defining steganographic signatures. This does not mean that we have abandoned the task of discovering the perpetrators. However, this potentially time-consuming task may be best done off-line after the necessary modifications have been made to prevent the covert channels, but before the original connection state is purged from our warden.

3 Related Work

The terms *covert channel* and *subliminal channel* are often used interchangeably, but in this paper we use the following definitions from the literature. Lampson defines a covert channel as a channel that is not intended for information transfer [20, 9]. In

[31], Simmons describes a subliminal channel as one where hidden data piggybacks on an innocuous-looking legitimate communication. By definition, steganographic carriers are subliminal channels since the communication appears to be innocent, but really has ulterior information embedded below the threshold of perception.

Active wardens have been an area of postulation since Simmons [31] introduced the *Prisoners' Problem* in 1983. Simmons presents Alice and Bob as prisoners that collectively wish to plan their escape. However, since they are in separate areas of the prison, all of their communication must pass through the warden, Willy. If Willy sees any attempts at secret communication in their messages, he will stymie their efforts by not allowing them to communicate in the future. Thus, Alice and Bob must use a subliminal channel to communicate their escape plan without alerting Willy. Since Willy knows that Alice and Bob may wish to communicate secretly, he must carefully analyze all correspondence between Alice and Bob, but he must do so without perceptively altering their message or incurring a noticeable time delay. In this context, Simmons defined a subliminal channel as a communications channel whose very existence is undetectable to a warden.

Active wardens have been discussed on several occasions [3, 1, 31, 4, 15] to actively block the creation of subliminal channels, but to date, there have been no published implementations of this type of warden. Meanwhile, firewalls are a routinely used form of active warden that is targeted at blocking unauthorized network access.

In [3], Anderson discusses both passive wardens, which monitor traffic and report when some unauthorized traffic is detected, and active wardens, who try to remove any information that could possibly be embedded in traffic that passes by. In [3], Anderson shows that there are methods 'more contrived than practical' where embedded data could survive a pass through an active warden.

In [6], Ettinger develops the idea of *critical distortion* in an active warden scenario between two game players, a data hider and a data attacker. Equilibria for the game is achieved when the communication channel is distorted to a level where covert channels will not survive. Ettinger observed that due to the large number of bits that both the data hider and the data attacker could modify, this problem was extremely complex. While we don't dispute this fact, our approach fundamentally differs from his in that Ettinger attempted to determine the critical distortion dynamically, without any prior knowledge of the steganographic carrier. Our technique implements static rule sets for a given carrier that are applied to the data as it traverses the network. By restricting the problem in this fashion, we are able to successfully eliminate steganography from certain carriers in Internet traffic.

In 1997, Petitcolas published *Stirmark* [26, 25, 34], which has some of the functionality of a warden, but does not automatically change all network information as it traverses a network. Instead, *Stirmark* is an application program that will attempt to remove steganography in a given image. If modified, *Stirmark* could be used as a networked warden for certain types of unstructured carriers. In contrast, our contributions in this paper focus primarily on structured carriers such as TCP/IP.

Also in the area of unstructured carriers, Johnson [12] tested several contemporary steganographic systems for robustness. His tests involved embedding information into an image, and then testing its survivability against a myriad of techniques including

format translation, bit-density translation, blurring, smoothing, adding and removing noise, edge-sharpening, rotation, and dilation. Johnson noted that tools that rely of bit-wise embedding methods failed all of the tests.

Digital watermarking [25, 13] uses many of the same techniques as steganography, but sometimes with an emphasis on robustness more than secrecy. Watermarks are designed to be tolerant of attempts to remove them by altering or transforming the carrier. An active warden would have a more difficult time removing a good watermark, but the detection of that watermark may also be proportionately easier.

A network intrusion detection system is a form of passive warden that observes network traffic in search of malicious attacks. However, there have been several studies of ways to subvert intrusion detection systems using techniques known as packet evasion [29, 24] which exploit ambiguities in the semantics of network protocols and differences in perspective between intrusion detection systems and end hosts. Recently, it has been shown that this kind of attack can be defended against through the use of a protocol scrubber [21] or a traffic normalizer [10] which reduces ambiguous traffic to a canonical form that can be more reliably monitored. Similar techniques have been used to limit the amount of information leaked to a system fingerprinting mechanism such as *nmap* [32]. While some of the mechanisms used to perform scrubbing and normalization are similar to that of an active warden, the problem domains differ.

4 Minimal Requisite Fidelity

Wardens have frequently been discussed as actors in a security system, but in our model, an active warden is a network service that is architecturally similar to a firewall, but functionally quite different. Like a firewall, a warden implements a site's security policy. To prevent attacks from the outside, inside, or both, the warden modifies all traffic to remove many, if not all, of the carriers that can be used for covert channels, subliminal channels, intrusion detection evasion, and even some forms of attacks. Because this warden is a network service, it must be concerned not only with the application data that it handles, but also with the network protocols used to exchange data.

One way to prevent the use of covert channels and subliminal channels across a network is to drastically alter all data that passes across that network and that may be used as a carrier. For example, if it is believed that data is embedded in color detail, all images can be converted to monochrome. However, this level of modification would disrupt users and is not generally acceptable.

An alternate technique for preventing the successful use of covert channels is to distort potential carriers just enough that any covert and subliminal channels in those carriers become unusable. If done carefully, the overt users of the carriers remain unaware of these modifications. We describe this modification of traffic as imposing *Minimal Requisite Fidelity*. This term captures the essence of both the opportunity for data embedding and a warden's defense. The basic premise is that for any communication there is some fidelity at which the data is interpreted by the recipient. For example, an image displayed in a web browser is intended for human consumption and need not possess any more information than is apparent to a human eye viewing a computer screen. However, the transmitted data may contain more detailed information than is

perceptible to the viewer. As described in the following section, minute differences in color, textures, saturation, or other measures can be used to hide a wealth of information. The paradigm of Minimal Requisite Fidelity refers to determining the threshold of fidelity that is required for overt communications with the recipient and then limiting the fidelity of network transmissions so that no additional information is preserved.

Since MRF preserves functionality while altering the exact values seen by the receiver, it makes the job of an attacker much more difficult, if not impossible. In this regard, an active warden enforcing MRF is very much like a network proxy. Such a warden acts as a *semantic proxy* by relaying the semantics of the protocol while insulating each end-point from the specific syntax created by that end-point. To date, there has been no theory behind proxies, but MRF could be used to define one.

The ability to perform this fidelity modification varies with the type of carrier being used. In the next section, we break carriers into two broad classes of *structured* and *unstructured* carriers and provide examples of how the Minimal Requisite Fidelity paradigm can be applied to them. We will show that the paradigm is equally applicable, but that additional constraints present with structured carriers allow for much stronger guarantees to be made.

5 Carrier Taxonomy

In this section, we will examine techniques for embedding data in some common examples of unstructured carriers and structured carriers. The definition of MRF for the two different types of carriers is quite different.

5.1 Unstructured Carriers

A subliminal channel is based on modifying a carrier in imperceptible ways. For what we call *unstructured carriers*, the limits to what can be changed are defined by fuzzy notions such as perception. Perception can be quantified and carriers can be subjected to statistical analysis, but there is no universal, objective bound to how much information can be altered for purposes of embedding. Below this level of perception, arbitrary changes can be made to the data in order to embed information. However, an active warden can make use of the exact same freedoms to destroy any embedded information.

Examples of techniques to embed data in unstructured carriers are *Null Ciphers* - hiding data in plain text [18]; *Least-Significant Bit Embedding* - modifying the least-significant bit of specified pixels that result in color variations that are not distinguishable to the human eye [14]; *Bit-Plane Embedding* - identifying noisy regions of each bit-plane in an image and replacing those regions with embedded data [19]; and *Discrete Cosine Transformation* - modifying and converting pixel values into frequency values using the IDCT [13].

Quantifying MRF for Unstructured Carriers: In each of these examples of unstructured carriers, a Minimal Requisite Fidelity can be defined. This would be the minimum amount of purity in an unstructured carrier that is needed to convey the meaning of the carrier. In the example of an image, this MRF would be the set of minimal colors that displays the image as seen by the human eye. In a null cipher, the MRF could be

achieved by slightly rewording phrases and adding spaces and tabs to the end of lines so the same meaning is conveyed, but in a slightly different format.

However, finding the correct Minimal Requisite Fidelity for unstructured carriers is challenging. Because there are not objective bounds to the carrier, a threshold of requisite fidelity must be chosen subjectively. This threshold can be based upon knowledge of human perception, or the typical use of data. However, there remains the possibility that a determined adversary will risk making perceptible changes for the sake of getting a signal through. For instance, a warden may thwart BPCS by modifying all noisy regions in an image, but the threshold for defining a noisy region is arbitrary. An adversary could embed data in less noisy regions at the expense of making them appear grainy. While a warden might not be able to make all images grainy, grainy images might legitimately occur and be let through. Nonetheless, a warden may be able to assume that preserving graininess is not a requirement. In this case, smoothing or randomizing of grain could be employed.

Clearly, there is a cycle of measure and counter-measure to this game. However, any time a warden can afford to reduce the fidelity of the carrier, the adversary's job gets harder. While this cycle may be arduous, it at least makes forward progress towards security.

5.2 Structured Carriers

In contrast to unstructured carriers such as plain text, *structured* carriers are instantiations of some well-defined syntax and semantics. In this section we focus on a significant example of the structured carrier, network protocols. We first present an example of network traffic embedding and then examine how this technique exploits the syntax and semantics of the protocol. This examination leads to a formal expression of Minimal Requisite Fidelity. The ability to make such formal expressions is a unique characteristic of structured protocols and enables wardens to more thoroughly apply the concept of Minimal Requisite Fidelity.

Network protocols such as the TCP/IP family of Internet protocols define both a syntax for network packets as well as the semantics used by systems exchanging packets. The syntax is the data format for packets that traverse the network. This syntax is not unlike the image encoding format of some unstructured carriers. What makes structured carriers different is the additional specification of semantics that describe how a packet is interpreted and what actions the end host will make based upon that packet.

For example, the *Covert TCP* [30] program manipulates TCP/IP header information to encode ASCII values in header fields. *Covert TCP* makes use of the fact that IP uses arbitrarily assigned numbers to identify packets. Each packet has an ID field containing a 16-bit number. This ID has no notion of order and is used purely to let a packet be fragmented while allowing the receiver to identify related fragments and reassemble the larger packet. Every associated fragment will contain the same ID, while fragments of different packets will contain different IDs. *Covert TCP* chooses IDs that contain data to be sent. As a simplified example, the string 'STEG' can be embedded in a series of four packets where the first packet has an ID equal to the ASCII value of 'S', the second has an ID equal to 'T' and so on.

Because the semantics of the ID field are so clearly defined, *Covert TCP* is able to fully exploit the protocol without the risk that its choice of ID numbers will cause changes that are perceptible to the recipients of the packet. However, an active warden can use the same fact to renumber IDs to thwart such channels. In the following sections, we provide a concrete analysis of the semantics of this example and how MRF can be absolutely applied to this type of carrier.

Quantifying MRF for Structured Carriers: Information theory provides a basis for analyzing the fidelity required to support the semantics of structured carriers. While the identifier field is not required to be a random variable, the difference between the amount of information contained in the field, 1 of 2^{16} values, and the amount of information provided to the receiver is startling. The receiver need only match a fragment to $1/n$ packets where n is the number of packets that the receiver may be reassembling at any point in time. For TCP, which accounts for the vast majority of traffic,[1] the value of n is bounded by the receiver's advertised window size and is typically zero since most upper-layer protocols tend to avoid fragmentation for performance reasons. Thus, in the typical case, the amount of entropy present in the identifier is much greater than the amount required by the protocol semantics.

This extra entropy can be used by programs such as *Covert TCP* or more sophisticated steganography in order to create a covert or subliminal channel. However, our definition of the amount of entropy required by the protocol semantics also leads us to search for a bijective transformation that randomizes this extra information while preserving semantics. With such a transformation, a warden can randomly permute the identifiers chosen by untrusted end systems.

Assuming that a warden used some permutation function, $f(x)$, an attacker could potentially learn the values of the renumbered packets and attempt to engineer an inverse function so that she may transmit packets with an inverted ID, $f^{-1}(x)$, that, when transformed by the warden, becomes the intended value, $f(f^{-1}(x)) = x$. However, this kind of security feature is the very problem that encryption systems address. Therefore, we can employ an encryption algorithm to perform this permutation. Thus, we can recast the problem of randomizing excess entropy as a solved problem of encrypting the packet field.

Covert TCP provides us with an example second problem with slightly different semantics. This example will exercise our reasoning and shows that our method for enforcing Minimal Requisite Fidelity has promise for additional carriers. *Covert TCP* can also embed data in the TCP initial sequence number, which is another arbitrarily chosen number. However, the semantics of this number are somewhat different in that subsequent packets from the initial sender will contain a sequence number computed by incrementing the initial sequence number. Further, the receiver will acknowledge the receipt of these sequence numbers back to the sender. Thus, the permutation must be applied only to the initial sequence number of a connection. If the warden saves the difference between the original and modified initial sequence number, it can re-apply this offset to all subsequent packets in that connection.

[1] In our traces TCP accounts for 93% of the traffic. Figures vary, but this is not an unusual amount.

6 MRF Analysis of IP

Having seen that Minimal Requisite Fidelity can be precisely identified and manipulated in structured carriers, we now perform a more complete examination of the protocol headers and semantics in IP. We choose to look at IP because it has well-defined semantics and because without addressing IP, no Internet traffic can be considered completely protected. This case study will validate the applicability of the MRF model and demonstrate how a warden can provide some assurances about entire protocol layers. This analysis differs from previous work in [10] and [21] in that we are stopping the covert flow of data rather than attacks by a malicious outsider.

The MRF analysis of all IP fields is presented below as a taxonomy of field semantics. For the sake of brevity, we do not discuss the individual IP option fields which are rarely used and in general are quite open to modification by both adversaries and wardens.

Constant: (Version, Padding) These fields are effectively constants than cannot be changed without fundamentally changing the functions of the protocol. The version field specifies which version of the protocol is being used. Any value other than 4 (IP version 4) will cause the remainder of the packet to be interpreted with a completely different set of syntax and semantics. For instance, version 6 is the latest version, and while not widely supported, has similar, but slightly different definitions for syntax and semantics. An IPv4 packet cannot be turned into a valid IPv6 packet by simply changing the version number. For the sake of brevity, we assume IPv4 and term this field a constant. However, a more holistic analysis would examine all other versions of IP.

Free: (Type of Service, Don't Fragment Flag, Reserved Bits) These fields can hold arbitrary values while preserving the basic functionality of the protocol. Thus, wardens should modify these variables religiously. Due to nuances of these fields, we suggest that a warden not randomize them, but instead set them to safe defaults for the warden's network environment.

Note that this categorization of the type of service field may change if Differentiated Services [22] becomes widely deployed. However the Diff-Serv architecture assumes that this field will be administered according to local network policy and the warden may be a party to that policy.

Decreasing: (Time To Live) The time to live is a counter value that is decremented at each hop. When the time to live reaches zero, the packet is discarded. This causes packets in routing loops to eventually be dropped. The TTL can be changed, but in order to preserve the routing loop behavior, the new TTL should always be lower than the existing TTL. Note that decreasing the TTL will prevent *traceroute* from working properly since it depends on TTL values being decremented only once per hop.

Tokens: (Identification, Source Address) These fields, as described earlier, serve to correlate packets. The values themselves are arbitrarily chosen and can be mapped to different values, but this mapping must be stable across packets. Source address has some additional constraints in that it will be used to form reply and error messages. Thus, it must refer to the originator's address or the address of system willing to proxy these messages to the originator. Network Address Translation is a widely-deployed technology that rewrites source addresses on traffic passing through a gateway [5].

Derivative: (Header Length, Header Checksum) These fields are determined by other aspects of the header. The length is determined by the number of options included in the header, while the header checksum is computed from all other fields in the IP header (excluding payload). If the checksum alone is changed, the packet will be dropped in transit.

Fragmentation: (More Fragments Flag, Fragment Offset, Total Length) The maximum amount of data that can be sent is bounded by what the upper-layer protocol provides, but the IP layer has flexibility in how a payload is fragmented and sent. Fragments can be reassembled into a larger packet and then re-fragmented along different boundaries.

Dependent: (Destination Address, Protocol, Payload) These fields are determined by upper-layer protocols. In general, every value is legal, but the legality of any specific value is determined by the upper-layer using the protocol. As a result, neither an adversary nor a warden can directly alter these values without altering the behavior of the protocol. However, a warden operating at a higher layer should cause these fields to be changed wherever possible. For instance, an adversary or a warden could segment the upper-layer packets differently in order to embed or remove information in packet sizes.

The destination address cannot be changed by a warden. However, it does exhibit a property common to other fields in the dependent category. If an adversary is creating packets that do not contain legitimate data streams, arbitrary values can be chosen for these fields. For instance, a malicious party could generate traffic to incorrect or fictitious destinations knowing that the route to that address will cause the packets to traverse a link where a collaborating receiver can eavesdrop on traffic and observe the message. Additionally, an adversary could target specific machines on a subnet to sent covert messages.

Although the IP header is compactly designed, it is worth noting that other protocols have additional fields that are *predicated*. These fields are always present, but are unused in some circumstances. In situations where the protocol does not use these fields, they are essentially reserved bits. As such, they are exceptional opportunities for embedding and should be modified by wardens.

Of the six types of fields that we have defined, the application of Minimal Requisite Fidelity is most complicated for the *token* category, for which we have already described a solution. We have shown that, at least for the IP layer, MRF can be precisely defined and applied to each header field. Thus an active warden can give a level of assurance that IP headers are not being used for subliminal channels. In addition, this description of semantics sheds light on what kind of semantic detail is necessary to describe a structured carrier.

6.1 MRF Validation

The previous examination of IP was based on the protocol specification. However, the specification is not necessarily indicative of real use. For example, a protocol implementation may make use of reserved bits that have not been standardized. To validate our examination and determine which fields can be safely modified, we therefore performed several feasibility studies by analyzing network traffic from several sites. We

Protocol	Field	% Usage	Number Packets
IP	options present	0.0006091%	38358
IP	ToS reserved bits set	12.2%	768423616
IP	ToS precedence bit set	0.5254%	33090880
IP	ToS delay bit set	11.76%	33090880
IP	Don't fragment	0.03692%	2324986
TCP	no options	56.21%	3441988705
TCP	reserved bits set	0.007008%	429164
TCP	urgent bit set	0.0002275%	13934
TCP	urgent ptr set	0.02417%	1479773
TCP	MaxSeg option	3.395%	207894081
TCP	Window option	0.6625%	40570864
TCP	Bubba option	1.666e-06%	102
TCP	Skeeter option	1.911e-06%	117
UDP	all zero checksum	24.2%	32235386

Fig. 2. Partial field usage statistics from a month-long trace (over six billion packets).

present this information as a case study, and are not trying to make any observations about traffic composition as a whole on the Internet. The purpose of this study was to determine which fields in IP, TCP, UDP, and ICMP can be safely modified by our wardens without breaking any applications. Our final rule sets for IP, TCP, UDP, and ICMP will be dependent on our observations from these studies.

In Figure 2, we list part of the results that we observed during our analysis. In some instances, the results were surprising and showed several discrepancies. Why, for example, is TCP's urgent bit set in 13934 packets, but the urgent pointer is non-zero in 1479773 packets? Why were TCP's reserved bits set in 429164 packets? Who could be using the Bubba and Skeeter options[2]? While it is possible that these inconsistencies are due to faulty implementations of TCP or network errors, we must rule out the possibility that legitimate communication channels are using these fields for justifiable purposes before we modify these bits with an active warden.

In addition to determining legitimate uses of each field, we must also ensure that the correctness of the IP and TCP protocols and associated applications don't change. For example, changing the TTL without a proxy would break *traceroute*. Clearing the TCP reserved bits could interfere with *Explicit Congestion Notification*. For each rule that we define for each bit in the packet header, we will need to verify that a legitimate service will not be broken.

[2] Communications on the Internet History mailing list from Frank Kastenholz explained the original assignment of TCP options 16 and 17 to Stev Knowles of ftp Software, Inc. for Diffie-Hellman Key Exchange.

7 Proof-of-Concept Implementation

To demonstrate the techniques that we introduced in this paper, we have developed a proof-of-concept implementation of software that hides data in several TCP/IP fields and a warden that prevents those fields from being exploited. We implemented this warden as a module in the *System for Modular Analysis and Continuous Queries* [7], an extensible system for analyzing streaming data. To test the warden, we modified the *Covert TCP* application [30] to embed data using each of the following methods. To date, we enforce the following rules in IP and TCP:

1. If the TCP Reset flag is set, there should be no TCP payload. Recalculate the the IP packet length to make sure that the only payload is the TCP header.
2. The IP identification field can be used as a covert channel (as demonstrated by *Covert TCP*). Assign a new IP ID to packets.
3. The TCP Initial Sequence Number (ISN) can be used as a covert channel (as demonstrated by *Covert TCP*). Assign a new ISN at the beginning of a connection. Correct subsequent packets accordingly.
4. Reserved bits in TCP can be used as a covert channel. Zero these out.
5. If URG = 0, the urgent pointer is ignored and can be used to send covert data. Zero out the urgent pointer in this instance.
6. Bounds checking on the urgent pointer. The pointer is an offset of data in the payload. If the pointer is larger than the payload size, it is illegal and should be reset to 0 and the urgent flag removed.

Though this implementation only exercises a few rules, our experimental results are encouraging, showing minimal delay of the network traffic and the complete elimination of embedded data in these fields. A complete warden implementation for IP, TCP, UDP, and ICMP is in progress, and our preliminary results are very promising.

7.1 Implementation Issues

A key performance limitation with any sort of automated active warden or firewall is the amount of state that must be saved for each packet or connection. The algorithms described in Figure 1 can be efficiently implemented since they require minimal state.

To translate IP IDs, the warden can encrypt the original ID using some key. Because the mapping must be stable across multiple packets during a connection, we assume that a distinct pseudo-random key is used for each connection. Thus, the warden need only store one key for each connection. If the speed of the cipher itself is an issue, we assume that a cryptographic co-processor can be used.

As described earlier, the semantics of sequence numbers require that the warden only encrypt the initial sequence number and save the offset between the original and new values. As described above, the handling of subsequent packets requires only basic mathematical operations. It is worth noting that this sort of mapping of sequence numbers is already supported by many Layer 7 switches that splice together separate TCP connections to the client and server.

8 Concluding Remarks

In this paper, we presented and discussed the paradigm of proactively preventing steganography, covert channels, and other forms of network attack. This paradigm uses a notion of *Minimal Requisite Fidelity* (MRF) to define the level of signal perturbation that is both acceptable to users and destructive to steganography. To develop the idea of MRF, we introduced the concepts of *unstructured* and *structured* carriers and gave several examples of how an attacker can exploit the use of anything more than the minimal fidelity that is required for overt communications. For structured carriers, we were able to take the analysis a step further and examine the feasibility of an active warden that rewrites all network packets to remove the opportunity for covert channels and steganography at the IP layer.

These initial explorations show a paradigm and a model with great promise. However, much work remains to define Minimal Requisite Fidelities for other carriers, and to integrate this model in with traditional layered security models. Wardens won't stop every form of attack, but if part of a more comprehensive security model for a site can greatly reduce the bandwidth of these attacks.

In addition to the techniques that we presented, there are additional dimensions of fidelity, such as timing, that must also be examined. As we have demonstrated, defining an objective Minimal Requisite Fidelity for unstructured carriers is a difficult problem, but one not without hope. For structured carriers such as network protocols, we believe that much more precise definitions of fidelity can be made and enforced through detailed analysis of protocol semantics.

Excitingly, we have found that this paradigm transcends specific categories such as steganography, network intrusions, and covert channels. The development of this paradigm has been a stimulating synthesis of experience in each of these areas, and, as such, we believe that the deployment of active wardens is a necessary addition to site security perimeters. Technologies such as active wardens are a new opportunity to create bi-directional security perimeters that protect against the malicious insider as well as the outside attacker.

References

1. R. J. Anderson, "Stretching the limits of steganography," *Springer Lecture Notes in Computer Science*, pp. 39–48, 1996, Special Issue on Information Hiding.
2. R. J. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, John Wiley and Sons, New York, New York, USA, 2001.
3. R. J. Anderson and F. A. P. Petitcolas, "On the limits of steganography," *IEEE Journal of Selected Areas in Communications*, vol. 16, no. 4, pp. 474–481, May 1998, Special Issue on copyright and privacy protection.
4. S. Craver, "On public-key steganography in the presence of an active warden," in *Proceedings of the Second Information Hiding Workshop*, Apr. 1998.
5. K. Egevang and P. Francis, "RFC 1631: The IP network address translator (NAT)," May 1994.
6. M. Ettinger, "Steganalysis and game equilibria," in *Information Hiding*, 1998, pp. 319–328.
7. M. Fisk and G. Varghese, "Agile and scalable analysis of network events," in *Proceedings of the SIGCOMM Internet Measurement Workshop*. ACM, Nov. 2002.

8. J. Fridrich, R. Du, and M. Long, "Steganalysis of LSB encoding in color images," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Aug. 2000.
9. V. D. Gilgor, "A guide to understanding covert channel analysis of trusted systems," Tech. Rep., National Computer Security Center, U.S. Department of Defense, 1993.
10. M. Handley, C. Kreibich, and V. Paxson, "Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics," in *Proceedings of USENIX Security Symposium*, 2001.
11. A. Havill, *The Spy Who Stayed Out In The Cold: The Secret Life of Double Agent Robert Hanssen*, St. Martin's Press, 2001.
12. N. F. Johnson, "Steganalysis of images created using current steganographic software," in *Proceedings of the Second Information Hiding Workshop*, Apr. 1998.
13. N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures*, Kluwer Academic Publishers, 2000.
14. N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *IEEE Computer*, pp. 26–34, Feb. 1998.
15. N. F. Johnson and S. Jajodia, "Steganalysis: The investigation of hidden information," in *Proceedings of the IEEE Information Technology Conference*, Sept. 1998.
16. "JP Hide and Seek," http://linux01.gwdg.de/ alatham/stego.html.
17. "JSteg Shell," http://www.tiac.net/users/korejwa/jsteg.htm.
18. D. Kahn, *The Codebreakers - The Story of Secret Writing*, Scribner, New York, New York, USA, 1996.
19. E. Kawaguchi and R. O. Eason, "Principle and applications of BPCS steganography," in *Proceedings of SPIE's International Symposium on Voice, Video, and Data Communications*, Nov. 1998.
20. B. W. Lampson, "A note on the confinement problem," *Communications of the ACM*, vol. 16, no. 10, pp. 613–615, 1973.
21. G. R. Malan, D. Watson, and F. Jahanian, "Transport and application protocol scrubbing," in *Proceedings of IEEE InfoCom*, Mar. 2000.
22. K. Nichols, S. Blake, F. Baker, and D. Black, "RFC 2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 headers," Dec. 1998.
23. "Outguess," http://www.outguess.org/.
24. V. Paxson, "Bro: A system for detecting network intruders in real-time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435–2463, Dec. 1999.
25. F. A. P. Petitcolas, "Watermarking schemes evaluation," *I.E.E.E. Signal Processing*, vol. 17, pp. 58–64, 2000.
26. F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in *Proceedings of Information Hiding, Second International Workshop, IH'98*, 1998.
27. S. Pluta, "United States of America vs. Robert P. Hanssen," http://www.fas.org/irp/ops/ci/hanssen_affidavit.html.
28. N. Provos and P. Honeyman, "Detecting steganographic content on the internet," in *Proceedings of the Network and Distributed Systems Security Symposium (NDSS)*, 2002.
29. T. H. Ptacek and T. N. Newsham, "Insertion, evasion, and denial of service: Eluding network intrusion detection," Tech. Rep., Secure Networks Inc., Jan. 1998.
30. C. H. Rowland, "Covert channels in the TCP/IP protocol suite," *First Monday*, 1996.
31. G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptography: Proceedings of Crypto-83*, D. Chaum, Ed. Aug. 1983, pp. 51–67, Plenum Press, New York and London, 1984.
32. M. Smart, G. R. Malan, and F. Jahanian, "Defeating TCP/IP stack fingerprinting," in *Proceedings of the 9th USENIX Security Symposium*, Aug. 2000.

33. “Stegdetect,” http://freshmeat.net/projects/stegdtetect/.
34. “Stirmark,” http://www.cl.cam.ac.uk/ fapp2/software/.