Technical Report MS-CIS-05-24

The Eavesdropper's Dilemma

Eric Cronin, Micah Sherr, and Matt Blaze
Distributed Systems Lab
Department of Computer and Information Science
University of Pennsylvania
{ecronin,msherr,blaze}@cis.upenn.edu

Abstract

This paper examines the problem of surreptitious Internet interception from the eavesdropper's point of view. We introduce the notion of 'fidelity" in digital eavesdropping. In particular, we formalize several kinds of "network noise" that might degrade fidelity, most notably "confusion," and show that reliable network interception may not be as simple as previously thought or even always possible. Finally, we suggest requirements for "high fidelity" network interception, and show how systems that do not meet these requirements can be vulnerable to countermeasures, which in some cases can be performed entirely by a third party without the cooperation or even knowledge of the communicating parties.

1 Introduction

Among the most basic simplifying assumptions of modern communications security is the notion that most communication channels should, by their very nature, be considered vulnerable to interception. It has long been considered almost reckless to suggest depending on any supposed intrinsic security properties of the network itself[1], and especially foolish in complex, decentralized, heterogeneously-controlled networks such as the modern Internet. Orthodox doctrine is that any security must be either provided end-to-end (as with cryptography), or not considered to exist at all.

While this rule-of-thumb well serves cautious confidential communicators, it is unsatisfying from the point of view of the *eavesdropper.* Paradoxically, while end-to-end security may be a prudent requirement for *assuring* confidentiality in most networks, it does not follow that a *lack* of end-to-end security automatically makes it possible to eavesdrop effectively.

In this paper, we investigate how the very properties that make it unwise to depend on the network for security can become a double edged sword that can threaten the eavesdropper at least as much as the communicator. We observe that while the Internet protocol stack and architecture make no confidentiality or authenticity guarantees regarding the traffic that passes across it, neither does it make any guarantees to those who seek to intercept this traffic (whether lawfully authorized to do so or not). While relatively accurate interception may be feasible on benign networks if performed with some care, we demonstrate that many of the most natural Internet eavesdropping configurations become inherently unreliable in the presence of active interference. Indeed, it is often both possible and feasible for an active antagonist – who may not even be a party to the communication – to artificially exacerbate the eavesdropper's problem, to the point of introducing ambiguities that make it unclear how to reconstruct the actual messages passed between targeted parties even when he cleartext can be captured accurately.

The central observation of this paper is that many of the foundations on which the Internet is based – a layered protocol stack, unreliable packet delivery, a lack of intrinsic authentication, etc – can work against the eavesdropper and can make high-assurance interception difficult or impossible under some circumstances. There are three main contributions to this paper. First, we provide a framework for considering "fidelity" in digital eavesdropping systems. Second, we formalize several kinds of "network noise" that might degrade fidelity, most notably "confusion" and show that reliable network interception may not be as simple as previously thought or even always possible. Finally, we suggest requirements for "high fidelity" network interception, and show how systems that do not meet these requirements can be vulnerable to countermeasures, which in some cases can performed entirely by a third party without the cooperation or even knowledge of the communicating parties.

[1] Quantum cryptography represents a respectable counterexample, of course.

1.1 Fidelity of digital eavesdropping

The concept of "fidelity" is usually associated with analog recording. Although the precise definition varies, it refers broadly to the degree to which a recording is free of noise and distortion and can render a realistic, "true to life" reproduction of the original. Even relatively simple analog audio recording systems must exercise care to achieve high fidelity, taking into account such factors as the limits of the recording media, appropriate sensitivity, amplification gain, microphone directionality, induced noise, copying degradation, and so on. A significant advantage to *digital* recording is the ability to make identical reproductions, free of additional distortion, once the analog to digital conversion has taken place.

It is therefore natural (although, as we shall see, often dangerously misguided) for a digital network eavesdropper to focus more on problems of *data capture* (ensuring that any bits sent by the sender to the receiver are reliably intercepted and recorded) and less on the reliability of the bits themselves. Unfortunately, however, complex networks introduce digital equivalents to analog noise and distortion that can significantly degrade passive interception even when it takes place entirely in the digital domain. To achieve high fidelity, a digital network eavesdropping system must balance complex tradeoffs that can be just as vexing as those arising in analog systems.

For the purposes of this paper, we assume that a network eavesdropper's goal is to reconstruct accurate transcripts of message streams (either between specific network points or to or from a particular source or sink) based on observed network traffic. Such eavesdropping might serve any of a range of (socially positive or negative) purposes, including law enforcement surveillance, intelligence gathering, malicious criminal activity, intrusion detection, network management or problem diagnosis. Regardless of their purpose, we consider here only *passive* eavesdroppers (i.e., those with the ability to monitor traffic traversing one or more links but without the ability to actively relay, interject, or alter data in transit) that are *surreptitious* (i.e., operating without the knowledge or cooperation of the monitored endpoints).

Let us introduce some terminology that will be used throughout the remainder of this paper.

As is customary, *Alice* and *Bob* will represent our network communicators; Alice will often be a source while Bob will be a sink (although, of course, in most protocols, the roles are symmetric and often alternating). *Eve* will be our eavesdropper.

An interception system is vulnerable to *evasion* if it does not capture and record in the stream all messages sent from Alice to Bob.

An interception system is vulnerable to *confusion* if it captures and records in the transcript messages *purportedly* from Alice to Bob but that are rejected or otherwise not processed by Bob.

An interception system is vulnerable to *obfuscation* if it cannot correctly interpret a message. Obfuscation might occur with respect to a message's content or its headers or both.

The *fidelity* of an interception system is its freedom from evasion, confusion and obfuscation.

Finally, a *countermeasure* is any deliberate technique that reduces the fidelity of an interception. The most familiar kind of eavesdropping countermeasure is encryption, which aims to achieve obfuscation, although note that end-to-end encryption obfuscates only the message contents, not the existence of the messages *per se.* A countermeasure might be *bilateral* (involving both Alice and Bob), *unilateral* (involving only Alice or only Bob), or *third party* (involving only a party other than Alice or Bob).

1.2 Related Work

There has been little prior work investigating the general problem of traffic interception from the eavesdropper's point of viewi [Bel00, BB00]. However, considerable research has addressed the related (but not identical) topic of information privacy. Cryptography, steganography, subliminal or covert channels [Sim83], winnowing and chaffing [Riv98], quantum communication [BBB$^+$90], and anonymous communications [DMS04, RR98], for example, all focus on establishing confidential communication. We look at the impact of these techniques on eavesdropping in Section 3.

Work from the eavesdropper's point of view has primarily been limited to the specialized area of intrusion detection [SP03, PP03, Pax99]. In a network intrusion detection system (NIDS), the primary goal of the listener (eavesdropper) is real-time analysis of incoming traffic to recognize attack signatures and detect anomalies. These systems are deployed at the borders of controlled networks where it becomes much easier to make assumptions about the machines within the network that the system protects. Additionally, the communication patterns of an attacker are also unique compared to general bidirectional communications (hence the NIDS is able to flag suspicious traffic). However, unlike a NIDS, a general purpose eavesdropper must process all traffic, both normal and anomalous. Because of these differences, we may draw from work on NIDS, but their applicability is limited by the different constraints on topology and communication characteristics.

Finally, we note that concern about eavesdropping interacts with the legal system in several ways. Appendix A looks at the legal aspects of this work in the framework of U.S. courts. Literature on computer forensics aimed at law

enforcement contains little mention of how to properly ensure that eavesdropping is done reliably [ecs02, Cas04].

2 The Eavesdropper's Dilemma: Ensuring Interception Fidelity

Assuming communication is not obfuscated, eavesdropping remains a nontrivial task. Alice can attempt to elude Eve through *evasion*, a process in which Alice constructs specially crafted messages that either fail to be detected by the electronic wiretap or are detected but subsequently ignored [PN98, SP03]. To reduce the possibility of evasion, Eve must exhibit adequate *sensitivity*. That is, Eve must intercept and consider all messages, even if they are seemingly irrelevant or anomalous.

In contrast, Eve should not incorrectly interpret noise as part of the communication. An adversary can induce noise through *confusion*, a novel eavesdropping countermeasure which we introduce in Section 3.2. If this noise forces Eve to consider multiple plausible interpretations of the data, then Eve's ability to make definitive statements regarding the conversation is reduced. Consequently, Eve must manifest sufficient *selectivity*. That is, she must be able to reliably differentiate between legitimate messages belonging to the conversation and noise.

To achieve high interception fidelity, Eve must therefore exhibit both adequate sensitivity and selectivity. We observe that these two requirements are inherently in conflict. Improved sensitivity results in a greater quantity of data that must be collected and processed, allowing an antagonist to more easily conduct a confusion attack. Likewise, increasing selectivity can lead to greater susceptibility to evasion by presenting the communicating parties with unmonitored channels in which they can safely converse. We term this predicament of seemingly contradictory requirements the *eavesdropper's dilemma*. Below, we explore this fundamental problem in greater detail.

2.1 Sensitivity

Eavesdroppers have limited monitoring capabilities. For example, an analog wiretap cannot discern all frequencies, nor can a digital eavesdropper interpret all possible message formats. Fortunately, specifications exist that restrict the syntax and semantics of communicated messages. Most eavesdropping systems rely on these standards to reduce the amount of information that must be captured and subsequently processed.

Although recording only the traffic that abides by specifications is useful in the absence of active countermeasures, this restricted interception affords Alice and Bob the opportunity to establish unmonitored channels (techniques for creating evasion are described in Section 3.1). In many cases, the evasion can be accomplished unilaterally, requiring only the sender's effort.

The consequences of a successful evasion attack are quite dire for the eavesdropper. An eavesdropper who is susceptible to evasion fails in its most basic task of data collection. No amount of post-interception analysis can reveal the contents of the lost communication.

Although existing work has examined the problem of evasion, it has done so only from the perspective of a NIDS [HKP01]. There, a *normalizer* intercepts messages and removes anomalies before signature checks are conducted. While normalization may be appropriate for a NIDS, it is unfortunately not well suited for eavesdropping. If normalization is passive (i.e., only Eve's interpretation is normalized), then her vulnerability to evasion *increases* since the normalization removes anomalies that Alice may have exploited to create her unmonitored channel. While active normalization reduces the likelihood of unmonitored channels by manipulating messages before they reach Bob, it does so at the expense of being detectable, making the technique inappropriate for clandestine eavesdropping operations.

We observe that the problem of providing adequate sensitivity is made even more difficult by the architecture of the Internet. The processing of Internet traffic relies on protocol layer abstractions. At each layer, a decision is made as to how to best interpret and summarize the input. For example, the physical layer in an 802.3 network translates frequencies, voltages, and amplitudes into bits. The physical attributes of the communication are then lost. The amount of information concerning a particular communication is thus diminished as the message progresses up the protocol stack. If Eve captures information at too high a level, then she risks susceptibility to evasion at all lower layers. Because she records at a higher layer, any misinterpretation made by Eve's hardware or software is uncorrectable.

To limit her exposure to evasion, Eve must maximize her sensitivity. She must not only record at the lowest possible layer, she must also consider all messages, even those that fall outside of some standard range of acceptability. Unfortunately, by exhibiting high sensitivity, she must then cope with the problem of selectivity.

2.2 Selectivity

To produce meaningful reconstructions of intercepted traffic, Eve must be able to eliminate noise from her transcripts. If Eve fails in this task and interprets a substantial quantity of noise, then the true messages may be indiscernible. In

order to be sufficiently selective, we propose that Eve must be capable of making three distinctions:

First, Eve must be able to determine whether each intercepted message originated from Alice. The lack of authentication information in Internet packets makes this a particularly daunting task [Bel89]. Thus, Eve's position relative to Alice is crucial. If she can eavesdrop at a granularity sufficient to capture only the messages that leave Alice's interface, then she can safely ignore outside interference. However, if Eve lacks this ability, then she cannot differentiate between legitimate messages from Alice and forged messages designed to confuse her interpretations.

Second, Eve must also have the ability to determine which of the intercepted messages will be successfully delivered to Bob. If properly positioned, Eve may be able discern which messages are actually from Alice. However, she is still vulnerable to an eavesdropping countermeasure in which Alice transmits noise to mask her true messages. Since both the noise and the legitimate messages originate from Alice, authenticity is in itself not sufficient. As we discuss in Section 3.2.2, Alice can create specially crafted noise that is intercepted by Eve but subsequently dropped in the network. Hence, in order to determine whether a message will reach its destination, Eve requires precise knowledge of the routers along the path from Alice to Bob, including their locations and configurations.

Finally, Eve must also be cognizant of how a given message will be processed by the receiving party. As we show in Section 3, techniques exist in which a party can inject uncertainty by crafting noise that will be received but subsequently ignored by Bob. To prevent against this type of attack, Eve must therefore possess considerable information about Bob. For example, Bob's hardware, protocol stack implementation, and application configurations may all impact whether received messages are processed or ignored.

3 Interception Countermeasures: Reducing Interception Fidelity

At first, it may appear that the discussion of eavesdropping countermeasures begins and ends with encryption. If Alice wishes to ensure that her communications are not monitored, the strong guarantees provided by cryptography are very attractive compared to the weaker confidentiality of other countermeasures. An eavesdropper must take a broader view, however. The network might not be giving the eavesdropper an accurate picture of the traffic, and active countermeasures other than encryption might exacerbate this.

When both Alice and Bob actively want to preserve confidentiality, encryption or other bilateral techniques which we group as *obfuscation* are appropriate. However, there are situations where, while obfuscation may not be used, it is realistic to expect other countermeasures may be used. When the discovery of even the presence of communications is damaging, the use of *evasion* countermeasures, which prevent messages from showing up in the eavesdropper's transcripts at all, may be employed by Alice. If Bob is a neutral third party, unwilling to help Alice by supporting encryption (e.g. most web servers, instant messaging services, etc.), a unilateral technique such as evasion or *confusion* may be employed. Additionally, as confusion can be applied by a third party, it may be present even when neither Alice nor Bob employ countermeasures, either maliciously or in furtherance of policy.

Below, we examine evasion and confusion countermeasures in more detail.

3.1 Evasion

When eavesdropping countermeasures other than obfuscation are looked at, it is almost always evasion which is considered. For evasion to be possible, three criteria must be realized. First, Bob must have greater sensitivity than Eve. Otherwise, Eve will intercept a superset of the messages received by Bob (such a situation results in susceptibility to confusion, as we discuss in Section 3.2), eliminating the possibility of unmonitored channels. Second, to craft messages that are received by Bob but ignored by the eavesdropper, she must know the precise levels that are tolerated by Bob and Eve. Finally, Alice must be capable of transmitting specialized messages that exploit these tolerances.

In the Internet architecture, there are many ways in which Alice can evade Eve. For example, ambiguities concerning the handling of overlapping IP fragments can lead to evasion [PN98]. To support varying maximum transmission units (MTUs) across the Internet, IP packets can be fragmented into smaller packets. It is the responsibility of the receiving host to reassemble the fragments and reconstruct the IP datagram. However, the IP standard does not specify how reconstruction should take place if two or more fragments overlap but contain different values for the overlapping portion [Pos81]. While some IP implementations consider the first arriving fragment, others do the reverse and use the last arriving fragment [Tim02, Pax99]. If Eve only records traffic at the transport layer, then she relies on her IP implementation to correctly handle any overlapping fragments. Hence, if Bob and Eve differ in how they interpret overlapping fragments, then Alice can evade Eve by sending confidential information in overlapping fragments. Evasion can also be done by exploiting ambiguities in TCP. For example, Alice can send packets with certain TCP flags enabled, causing certain stacks to discard the messages [PN98]. If Eve cannot process such messages, then this type of evasion may be feasible.

Although often effective, the techniques just described do not represent fundamental limitations and instead rely on weaknesses in eavesdropping systems. If an eavesdropping system is replaced with one that offers greater sensitivity, then Eve may be able to detect the evasion.

A more powerful evasion technique is possible at the physical layer when all parties share the same communication medium (e.g., an Ethernet bus or an 802.11 network). At the physical layer, standards exist that define acceptable ranges for amplitudes, frequencies, voltages, and so forth. Network devices, particularly commodity hardware, do not strictly abide by these standards and often interpret messages sent outside of the specified ranges. To evade the eavesdropper, Alice can transmit messages at a frequency, amplitude, or voltage that is imperceivable to Eve but acceptable by Bob. (Note that this type of physical evasion is more difficult when Alice, Bob, and Eve do not share a communication medium, as intermediary routers act as normalizers and reduce the likelihood of an effective evasion attack.) Generally, if Eve is less sensitive than Bob and the three parties share a communication medium, then Eve is susceptible to evasion. As we will show in the following section, Eve's obvious counter-countermeasure (i.e., enhancing her sensitivity) has the unfortunate effect of increasing her vulnerability to confusion.

3.2 Confusion

In this section, we introduce *confusion*, a novel eavesdropping countermeasure.[2] Unlike obfuscation and evasion in which the goal is to conceal traffic from the eavesdropper, confusion attempts to reduce Eve's interception fidelity by masking the true message in a deluge of noise. Like evasion, it is a unidirectional and unilateral countermeasure and is completely transparent to the receiver, but it targets selectivity and not sensitivity.

At its weakest, confusion seeks to achieve *deniability*, providing Alice with the opportunity to plausibly claim that Eve's view of the communication is ungenuine or inaccurate. At its strongest, confusion provides sufficient cover to make it difficult or impossible for Eve to derive any meaning from the interception at all. (Even relatively weak deniability may be sufficient to thwart certain kinds of legal interceptions; see Appendix A for discussion.)

Because it can be a third-party countermeasure, it is useful to introduce a new principal called a *confuser*, which in some cases may be Alice or Bob. In addition, because confusion introduces noise into the network, there must be a way to remove it, using a *noise filter*. If Eve lacks sufficient selectivity, she will be unable to differentiate Alice's traffic from the noise, diminishing her interception fidelity.

To ensure reliable communication, the noise injected by the confuser should not be interpreted by Bob. Located somewhere along the path from Alice to Bob, the noise filter intercepts and removes the noise, allowing only the legitimate messages to be processed by Bob. The noise filter could be an active third-party participant, an aspect of the network, or a component of Bob's system. Regardless of its implementation, the noise filter should remove noise after Eve conducts her interception and in a manner that is transparent to Bob.

One requisite of confusion is that there must be an asymmetry in knowledge between the confuser and Eve. The confuser must have sufficient comprehension regarding the behavior of the noise filter that it can generate noise that it knows will be filtered before reaching Bob. For confusion to be effective, Eve cannot also yield this knowledge, else she too can filter the noise and remove all ambiguity.

The confuser must also be cognizant of channels that are associated with but differ from the principal communication stream. Internet traffic is often multi-faceted and bidirectional: TCP utilizes a backchannel for control data, DNS lookups often accompany URL requests, and so forth. If forged noise results in irregularities in these associated streams, then the eavesdropper can use them to disambiguate the noise and counter confusion. Fortunately for the confuser, these associated channels are rarely authenticated and are themselves subject to confusion (for example, by injecting superfluous TCP RSTs, DNS lookups, ICMP TTL-exceeded messages, etc.). Thus, not only must a confuser inject noise to create ambiguity in the message stream, it must also ensure that all related traffic is also confused.

We note that, although similar, confusion differs from *insertion*, a technique used to bypass a NIDS through the injection of extraneous data [PN98]. There, the goal is to thwart signature checking by causing the NIDS to misread the communication. With insertion, there is no uncertainty as to the originator of the messages. Alice is the perpetrator, and no messages are spoofed. In contrast, confusion can often be applied by a third-party who does not other participate in the communication. Additionally, unlike insertion, confusion does not depend on the intercepter choosing one particular and incorrect interpretation. Rather, confusion uses noise injection to create multiple possible interpretations of a communication stream. The goal is not misinterpretation *per se*, but rather the causation of ambiguity and uncertainty.

Although we do not advocate that confusion be used as a general confidentiality technique, we briefly note that confusion has some interesting qualities that make it particularly attractive as an eavesdropping countermeasure.

[2] We first raised the possibility of confusion in a not-yet-published position paper [CSB05].

- While cryptography is typically used in a manner that ensures the confidentiality only of message payloads, confusion protects both a message's contents and metadata. It may therefore be advantageous to combine confusion with encryption to mask signaling information as well as content.
- Since confusion is transparent to Bob, it may be easily incorporated into existing protocols. Thus, it may be particularly useful when legacy applications and protocols cannot be easily upgraded or replaced.
- If both the confuser and the noise filter are third-parties, then neither Alice nor Bob needs to be aware of the confusion. Unlike obfuscation in which it is obvious that Alice and Bob have colluded to disguise their messages, confusion allows Alice and Bob to deny that they even attempted to communicate privately.
- Unlike cryptography, confusion does not require any computationally expensive operations. Confusion is therefore well suited for situations in which more traditional confidentiality measures are impractical (for example, low-power wireless sensor networks).

3.2.1 Semantic Noise Generation

Confusion has succeeded at reducing Eve's interception fidelity if she must consider numerous *plausible* interpretations. However, noise cannot be generated arbitrarily. Rather, noise must form realistic communications. For example, if Eve must choose between two interpretations of an email message – "meet me at four" versus a nonsensical string of random characters – she can with reasonable confidence select the former and discard the latter.

If Alice functions as both the sender and the confuser, then semantically valid noise generation is quite straightforward. Since she is aware of her true messages, she can craft a number of false messages (noise) to cover her true meaning (e.g., "meet me at three", "meet me at five", etc.). In some cases, Alice may even be able to achieve confidentiality by transmitting each byte of her message in a separate TCP packet. For each byte of her legitimate message, Alice then sends the 255 remaining octets as single-byte noise messages. If Eve has poor selectivity and each possible byte value is equally likely, then confusion provides perfect secrecy. However, assuming an m-byte message, this would require $(256 \times m)$ TCP packets, so the communication cost associated with this technique may be prohibitive in certain topologies.

There are, however, more practical approaches for inserting ambiguity. Most protocols limit syntaxes and message contents. It is therefore gratuitous to generate noise that corresponds to impossible interpretations. Instead, a confuser can focus his efforts on producing more convincing noise. For example, a confuser can produce valid HTTP requests with URLs that differ from those requested by Alice (the Confusing Wireless Access Point described in Section 4.2.1 uses such a technique). Although confidentiality is not guaranteed, Eve may be unable to identify Alice's true web requests provided that the confuser can produce a sufficient volume of noise. Even if Eve has the capacity to enumerate possible interpretations, her uncertainty affords Alice some measure of deniability.

Finally, confusion is still possible for protocols such as instant messaging and SMTP that transport written text. In such cases, a confuser can utilize natural language generators to produce plausible communications. Notable examples of these generators include *dadadodo* [Zaw03], a Markov-Model based tool that produces novel text similar in style to a provided corpus, and SCIgen [sci], a tool for generating academic computer science papers[3]. Although noise generation is more computationally complex for these protocols, it is not infeasible.

3.2.2 Confusion in the Internet Architecture

By design, the Internet is a very heterogeneous system. Machines of differing hardware and software configurations communicate and interoperate through the use of standard protocols. However, ambiguities in implementations, configurations, and protocol specifications create the opportunity for non-uniformity in the processing of specially crafted messages. Confusion exploits these inconsistencies by forcing the eavesdropper to consider multiple plausible interpretations of its transcripts. The IP and TCP specifications (which famously advise "be conservative in what you do, be liberal in what you accept from others. [Pos81]") thus aggravate the problem of proper selectivity by recommending that implementations accept even outlier communications.

Below, we explore various vectors and techniques for injecting confusion in the Internet architecture. The confusion countermeasures are not intended to be exhaustive; rather, their purpose is to illustrate the ease and effectiveness at which reliable interception can be defeated.

[3]The ability of SCIgen to produce seemingly valid English text is best illustrated by the acceptance of one of its computer-generated articles [SAK05] in an academic conference.

At the Physical Layer Many of the evasion techniques discussed in Section 3.1 can be recast as confusion countermeasures. As depicted in Figure 1 (*left*), we assume a topology in which all parties share the same communication medium (e.g., a common bus or a wireless network). If Eve is more sensitive than Bob, then a third-party confuser can inject noise that is processed by Eve but ignored by Bob. As a result, Eve is forced to consider multiple interpretations, while Bob only sees the true message text.

Confusion is only possible if Eve has greater sensitivity than Bob. Otherwise, the confuser's noise is perceived by Bob, hindering reliable communication. Note that this is exactly the *opposite* criterion for susceptibility to evasion. As explained in Section 3.1, an eavesdropper is vulnerable to evasion if she is *less* sensitive than Bob. This set of contradictory requirements for reliable eavesdropping is a prime example of the eavesdropper's dilemma. While it may seem obvious that Eve can counter confusion by discarding outlier messages, doing so makes her more vulnerable to evasion. In contrast, increasing her sensitivity to avoid evasion makes Eve more susceptible to confusion. Hence, in eavesdropping configurations in which Alice, Bob, and Eve share a common communication medium, Eve's interception fidelity is inherently limited.

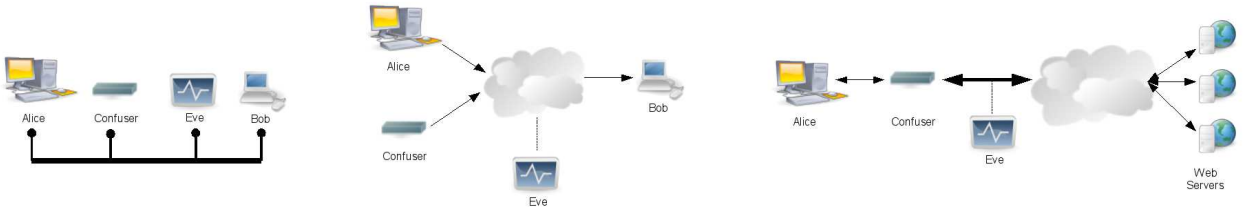

Figure 1: *Left:* An eavesdropping topology in which all parties communicate via the same shared bus. *Center:* A configuration in which Eve is located on the network between Alice and Bob. *Right:* A topology in which the confuser also functions as a router.

At the Link Layer Confusion is possible at the link layer if the confuser and Eve share the same Ethernet. A typical example of such a topology is an unencrypted 802.11 network in which Eve "sniffs" wireless transmissions. (If the confuser and Eve reside on separate networks, then the intermediary routers act as normalizers, making link-layer confusion very difficult.)

As we show empirically in Section 4, current eavesdropping systems suffer from inadequate selectivity. Although most eavesdropping systems are capable of recording traffic at the link layer, they often ignore Ethernet frames and instead process messages at either the network or transport layer. By crafting Ethernet frames with invalid MAC destination addresses, a confuser can inject noise that is processed by Eve but fails to be delivered to Bob [PN98]. Neither Bob nor the local gateway will process the noise since their operating systems silently discard Ethernet frames whose MAC addresses do not match that of the network interface.

This technique is obviously only effective when Eve has poor selectivity. If Eve examined the Ethernet frames, she would be capable of distinguishing the noise from the message text. Unlike other confusion countermeasures, the MAC technique is not indicative of a fundamental limitation of electronic eavesdropping. However, the significance of the approach is that it illustrates the dangers of inadequate selectivity: An eavesdropping system that fails to properly process Ethernet frames *is* inherently vulnerable to this form of confusion. Accordingly, an Internet eavesdropping system that observes traffic on a local Ethernet cannot claim to have high interception fidelity unless it both intercepts and processes link layer headers.

At the Network Layer If Eve intercepts a packet on the path from Alice and Bob (see Figure 1, *center*), she must carefully examine the packet's IP header to form an opinion as to whether the packet is deliverable. There are several reasons that a packet may fail to be delivered: the packet's checksum may be incorrect, IP options may be specified that are unsupported by an intermediary router (e.g., source routing), the packet's size may exceed a hop's MTU, or the initial time-to-live (TTL) value may be insufficient to reach Bob [Pos81, PN98]. If the confuser has more knowledge about the network than Eve, he can inject noise that will be dropped either before reaching Bob or by Bob's IP implementation. If Eve processes all intercepted IP packets (which, as we show in Section 4, is the case with all tested eavesdropping systems), then she will interpret the noise along with the legitimate traffic.

As with the link layer techniques, the network layer confusion countermeasures highlight weaknesses in current eavesdropping systems. By enhancing Eve's selectivity, many of these countermeasures can be eliminated. However, an eavesdropper that either does not examine IP headers or lacks sufficient selectivity to determine whether packets are deliverable is inherently vulnerable to this type of confusion.

Confuser-in-the-Middle A more fundamental limitation of reliable electronic eavesdropping occurs when a confuser can position itself between Alice and Eve (as shown in Figure 1, *right*). In this *confuser-in-the-middle* approach, Alice requests information from one or more services (e.g., web servers). Here, we make the conservative assumption that Eve has perfect sensitivity. Additionally, we require that the service is stateless (as is, for example, the case with most content-providing web sites) and does not utilize authorization. Although the confuser-in-the-middle technique does not guarantee confidentiality, it prevents Eve from definitely identifying Alice's requests from the noise.

In our topology, Alice and the confuser play multiple roles. Alice functions as both the sender and the receiver. She transmits her requests and receives their corresponding responses. In addition to generating noise, the confuser acts as a router and a noise filter. It receives requests from Alice and relays them to the next hop, and conversely, receives responses from the requested services and forwards them towards Alice.

The confuser inserts noise by forging requests from Alice to servers on the Internet. For example, if Alice wishes to browse the web, the confuser can forge thousands of HTTP requests to various sites on the Internet. The confuser then filters out the responses, allowing only traffic corresponding to Alice's true requests to be routed back to her.

Due to her location in the topology, Eve cannot differentiate Alice's messages from the forged noise. In fact, confuser-in-the-middle techniques have the interesting property that Eve cannot positively determine that confusion has even taken place. Moreover, since all messages may have originated from the confuser, Eve cannot reliably conclude that Alice transmitted *any* requests. In such an eavesdropping topology, any claims made by Eve concerning intercepted requests cannot be substantiated.

Semantic Confusion *Semantic confusion* refers to the class of confusion techniques in which some aspect of Bob's application acts as the noise filter. If the confuser has knowledge of Bob's configuration, he can exploit that information to generate noise that Bob will ignore.

An example of semantic confusion is *email chaff*, an eavesdropping countermeasure that uses email spam filters to induce confusion. A variety of techniques have been proposed that attempt to identify incoming spam messages [AKCS00, SDHH98, CM01, ZZY04, OV03]. Many of these approaches assign a "spam score" to incoming emails. Messages that have a score above an adjustable threshold are considered spam and are either deleted or moved to a special mail folder.

If the confuser has knowledge of both the classification mechanisms and thresholds used by Bob's spam filter, he can generate emails that will be caught by Bob's spam filter (for example, by inserting key phrases). Although Bob will see an increase in the number of messages he receives, the confuser's noise will be marked as spam and he can focus his attention on only the legitimate messages.

Email chaff does not ensure any significant measure of confidentiality. The contents of Alice's emails are sent in the clear and are easily monitored by an eavesdropper. However, email chaff does provide Alice with deniability. The presence of a third-party confuser can significantly increase the burden of an eavesdropper who must prove her intercepts are reliable. Upon being confronted with an intercepted email, Alice can plausibly argue that the message is a forgery from a third-party confuser. Moreover, if the quantity of noise is much greater than that of Alice's messages, Eve's ability to select only the legitimate messages may be significantly impaired.

Eve can do little to counter semantic confusion. She cannot ignore messages that she conjectures are spam, else she risks evasion. Unless Eve acquires acute knowledge of Bob's configuration, she cannot improve her selectivity, as messages and noise are only distinguishable by their semantic contents. In general, if semantic confusion is possible (as in the case with emails), Eve cannot maintain high interception fidelity.

4 Failure of Current Eavesdropping

Performing eavesdropping on digital networks is a two step process. First, the eavesdropper must select or construct an eavesdropping tool which sufficiently models the receiver. Then, the eavesdropper must select a convenient place in the network to attach this tool and gather information. In this section, we show through a series of experiments how there are serious vulnerabilities at both of these stages with the tools and practices commonly used today. We focused on evasion and confusion, as obfuscation has been explored in numerous other works.

4.1 Vulnerable Implementations

To demonstrate the susceptibility of current eavesdropping tools to confusion, we implemented the MAC and TTL confusion techniques described in Section 3.2.2 and originally introduced in [PN98]. (Fragroute [Son99] also provides an implementation of these techniques, but it is only suitable for NIDS attacks, not general purpose communication.) The MAC approach relies on generating noise with invalid MAC destination addresses. While Eve will process the

noise, the local gateway will not route such packets since it only accepts correctly addressed Ethernet frames. In the TTL technique, the confuser introduces noise with TTLs that are sufficient to reach Eve but not Bob. Note that both techniques can be trivially defeated by providing adequate selectivity. Here, our aim is not to introduce formidable countermeasures. Rather, we show that the current generation of eavesdropping tools are highly susceptible to even these weak forms of confusion.

In our experiments, Alice transmits an email via SMTP to our institution's email server (Bob). To confuse Eve, Alice (functioning as the confuser) injects spurious noise using either the MAC or the TTL confusion techniques. To maximize confusion, Alice sends both the legitimate email and the noise in byte-sized packets. For every byte of legitimate text, Alice sends 8 noise packets. Of the 8 noise streams, the first is comprised of a "cover message"[4]. The stream, although composed of noise, constitutes a false but sensical message (a passage from Dickens' "A Tale of Two Cities" [Dic59]). The remaining 7 streams of noise consist of random characters. In an attempt to cause Eve to interpret the false stream rather than her true message, Alice always sends the false stream first, followed by a random intermixing of the legitimate stream and the 7 random noise streams.

We tested our link and network layer confusion tools against 11 eavesdropping systems, ranging from commercial applications to free open-source toolkits (descriptions of the eavesdropping systems are provided in Appendix B). Experiments were conducted on a testbed network in which Alice and Eve reside on the same local subnet. From this subnet, a minimum TTL of 5 is required to reach Bob. Both Alice and Eve are Pentium servers with 3COM Fast EtherLink XL 100MB/s network cards and are connected via a 100MB/s switch.

	No Confusion (byte-sized pkts)		MAC Confusion		TTL Confusion	
Software	**Interpretation**	**Detected Anomalies**	**Interpretation**	**Detected Anomalies**	**Interpretation**	**Detected Anomalies**
bro	Success	None reported	Failure (Covertext)	Retransmission Inconsistency	Failure (Covertext)	Retransmission Inconsistency
chaosreader	Success	None Reported	Failure (Random noise)	None reported	Failure (Random noise)	None reported
CommView Eval. Version	Success	None reported	Failure (Covertext)	None reported	Failure (Covertext)	None reported
ethereal	Success	None reported	Failure (Covertext)	None reported	Failure (Covertext)	None reported
NetworkActiv PIAFCTM	Success	None reported	Failure (Covertext)	None reported	Failure (Covertext)	None reported
Sniffem	Failure (Random noise)	None reported	Failure (Random noise)	None reported	Failure (Random noise)	None reported
snort-replay	Success	None reported	Failure (Random noise)	None reported	Failure (Random noise)	None reported
snort-stream4	Success	None reported	Failure (Random Noise)	None reported	Failure (Random Noise)	TTL LIMIT Exceeded
tcpick	Success	None reported	Failure (Covertext)	None reported	Failure (Covertext)	None reported
tcptrace	Success	None reported	Failure (Random noise)	TCP DUPs detected	Failure (Random noise)	TCP DUPs detected
tcpflow	Success	None reported	Failure (Random noise)	None reported	Failure (Random noise)	None reported

Success - The eavesdropping application correctly interpreted the messagetext.
Failure (Covertext) - The eavesdropping application interpreted the covertext as the legitimate messagetext. See Figure 3.
Failure (Random noise) - No discernible English text could be obtained from this interpretation.

Figure 2: Ineffectiveness of various eavesdropping software against confusion techniques.

The performance of the eavesdroppers in the presence of confusion was startlingly lacking. Figure 2 describes Eve's (in)ability to reliably reconstruct the email messages. Although all but one eavesdropping packages were able to correctly reconstruct Alice's message in the absence of confusion, all tested systems failed to interpret her message once either confusion technique was applied. Anomalies were only reported by 18% of the eavesdroppers with the MAC-based approach and 27% when TTL confusion was used. Moreover, the cover message was perceived as the email in 45% of the cases when either technique was utilized (see Figure 3). In all cases, the email server (Bob) correctly received Alice's communication and delivered the email to its intended recipient.

[4] Although similar to steganography in which the messagetext is embedded but hidden within the covertext, our approach differs from steganography in that the covertext and the messagetext are sent as independent streams. Hence, the ability of confusion to provide covertext depends on Eve's method of reconstructing traffic and is independent of the content (e.g., video, audio, or text) of the communicated information.

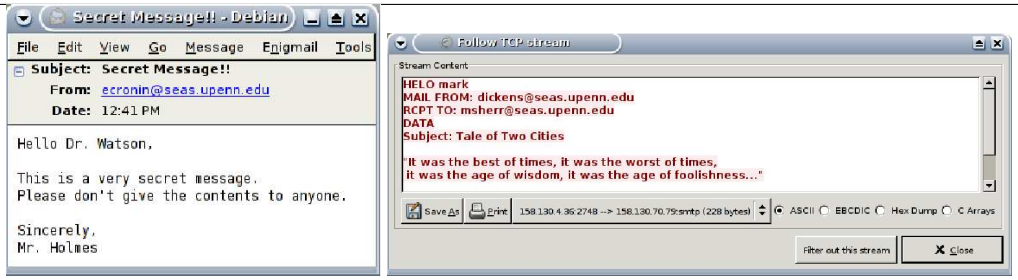

Figure 3: Secret message received by SMTP server and intended recipient (*left*). An eavesdropping system's (Ethereal) reconstruction in which the covertext is incorrectly interpreted as the legitimate message (*right*).

4.2 Vulnerable Positions

4.2.1 Injecting Confusion in 802.11 Networks

The confuser-in-the-middle countermeasure presented in Section 3.2 exposed a fundamental limitation of certain eavesdropping configurations due to the positioning of the eavesdropper. We apply this countermeasure in the creation of a *Confusing Access Point* (CAP).

With wired Ethernet, the widespread deployment of switches to replace hubs and older shared bus technologies has somewhat reduced the risk of malicious users passively eavesdropping on the local network segment. With wireless networks, however, the problem remains, and is in many ways worsened due to the unmanaged, public nature of many wireless networks. Even when authentication such as WEP or WPA is used to prevent an eavesdropper from joining a random network, the broadcast nature of radio communication makes all packets visible to any member of the wireless network. The obvious obfuscation technique of establishing pairwise keys between each host on the network and the access point is unsupported by existing wireless protocols. Moreover, enabling such pairwise encryption would require modifying both the access point as well as every client that connects to the network.

Confusion, however, has the advantage that it requires neither software modifications nor the establishment of pairwise keys. Confusion provides a technique for an access point to protect its connected hosts from local eavesdropping, including those hosts that may be unable or uninterested in encrypting their communications.

The CAP performs the standard functions of a wireless access point (AP), and to the clients of the network appears no different than an ordinary AP. In addition, a parallel process monitors connected hosts and fills the wireless network with forged traffic to and from these hosts. To an eavesdropper, the real and fake traffic is indistinguishable, and the fidelity of any intercepts is degraded.

Host	Actual # Connections	Observed # Connections
Host 1	470	1050
Host 2	160	691
Host 3	299	630

Figure 4: Actual number of connections and number of connections observed by Eve for Confusing Access Point.

Confusion is provided at the application layer. Entire connections are simulated, with the IP and MAC addresses forged to match clients on the wireless network. The CAP is implemented using Linux's built in networking for the access point operations (NAT, DHCP, routing, etc.) and a confusion daemon written using libnet [lib] and libdnet [Son]. Currently, only HTTP traffic is confused, although extending the system to protect other protocols is straightforward.

The confusion daemon has two tasks: to gather transcripts of legitimate looking connections and to replay these connections so that they appear to originate from one of the hosts on the wireless network. To gather realistic traffic, the confusion daemon uses the Google search engine to find random URLs. It then records the packets sent and received from requesting these URLs and all embedded images. To replay the streams, the daemon first rewrites the headers of the packets, replacing the IP and MAC addresses with those of the host being protected. The packets are then injected onto the wireless network. Since the protected host knows nothing about these sessions, it will generate TCP RST packets in response to this traffic. The CAP drops these RST packets instead of forwarding them to the web server. Additionally, the CAP constructs fake RST packets for any legitimate HTTP traffic destined to the host, making the two streams identical from the transport layer.

CAP was implemented on an IBM Thinkpad running Debian Linux with a PrismII chipset wireless card. Three clients were connected to the CAP for a duration of 20 minutes. During that time, each client engaged in "normal web browsing". Traffic was captured both at the CAP as well as on the three hosts. The results are presented in Figure 4.

The eavesdropper saw between two and four times as many connections as the clients actually generated. None of the clients reported noticing anything unusual about the network.

Because this technique emulates complete TCP sessions with correct IP and MAC addresses, the confused traffic is indistinguishable from legitimate traffic at the lower layers where current eavesdropping software operates. At the application layer, our simple cover traffic may not be completely indistinguishable to a trained human eavesdropper, so this technique is better suited to deniability rather than absolute confidentiality. However, the large number of sessions which must be evaluated requires some level of automation to reduce the number of possible sessions to a reasonable level.

4.2.2 POTS Evasion and Confusion

As we have seen so far, confusion and evasion can be practical threats to digital Internet eavesdropping, and indeed, such systems are the focus of this paper. However, similar techniques can also be applied to analog networks, especially when analog to digital conversion is performed. For a simple example, we consider voice telephone signaling between the subscriber and the switch.

Analog telephone service, also known as "plain old telephone service" (POTS), uses analog touch-tones to signal the caller's desired number. The touch-tone system is an international standard known as DTMF [Int88], followed by both telecommunications equipment manufacturers and the telephone network to ensure compatibility. Each of the 16 DTMF digits is comprised of two base frequencies. The specification lists acceptable ranges for tone duration, spacing, frequency, amplitude, and twist (relative amplitude of the high and low frequencies).

Constructing a decoder for DTMF signals which exactly follows these specifications turns out to be surprisingly difficult and expensive. Instead, most decoders relax the tolerances in one or more of the dimensions of the standard, accepting tones which are a little too loud or quiet, or a little out of pitch, etc. Since each decoder has a unique range of acceptable tones, it is possible to construct out-of-specification tones that some decoders will see but others will miss. Using this knowledge, it is possible to both evade and confuse an eavesdropper.

To test the practicality of this countermeasure, we constructed a simulated phone network in our lab. A Teltone TLS-5 Telephone Line Simulator was used as the central office, providing dialtone and four phone lines with unique phone numbers, as well as DTMF switching to connect calls between lines. Alice, the call initiator, was an Ameritec AM8a PCM/VF call analyzer with an ordinary POTS handset connected, Bob was a second POTS handset, and Eve was a Metrotel VNA 70a DTMF decoder.

Among its many features, the AM8a call analyzer used by Alice allows DTMF codes to be generated with very precise and configurable parameters. For our experiments, we focused on two variables: the frequency and amplitude of the higher of the two DTMF frequencies. Through binary search, settings which allowed for both evasion and confusion were discovered. Evasion can be applied by decreasing the amplitude of the higher frequency. At -39dBm, the central office still correctly decodes Alice's signal and completes the call, while the eavesdropper records nothing. Similarly, if instead the high frequency is increased by 3Hz the central office no longer recognizes Alice's touch-tones, but the eavesdropper records them as having been dialed. Using Alice's handset in coordination with the AM8a, the legitimate number can be dialed interspersed with out-of-range digits to provide confusion. In addition, although we did not test the scenario, by combining both techniques its is clear that Alice could drive Eve to a specific false phone number.

This experiment highlights the challenges which face an eavesdropper when positioned too close to the sender. Limited sensitivity and imperfect selectivity make it susceptible to both evasion and confusion countermeasures. While Eve may be certain that intercepts originate from Alice, she cannot be certain of where in the telephone network they terminate. A far more reliable form of dialed number recording is therefore achieved through analysis of call detail records generated by the switch itself, but this is, of course, not surreptitious with respect to the operators of the switch.

5 Improving Interception Fidelity

The experiments described in the previous section show how unilateral countermeasures can reduce the interception fidelity of eavesdropping systems. In this section, we explore methods to improve Eve's resilience to such countermeasures.

Enhancing Sensitivity To reduce her susceptibility to evasion, Eve can improve her sensitivity. This implies recording at the lowest possible OSI layer, and recording everything available (even data that appears to be erroneous). Any action that could have been performed automatically by lower layers, such as discarding corrupt packets, can be carefully emulated by Eve in a more selective manner.

Unfortunately, this advice may be hard to follow. For example, many authorized uses of eavesdropping in the United States operate under strict limitations on what can be recorded to prevent traffic of those not under suspicion from being observed (more information on this subject is provided in Appendix A). In such environments the steps Eve can take to improve sensitivity are reduced.

Enhancing Confusion Detection and Eavesdropper Selectivity In some situations, confusion may be made ineffective by deploying confusion-aware eavesdroppers. For example, the MAC confusion technique described in Section 3.2.2 can be defeated with improved software. By enhancing her sensitivity, Eve may be able to better identify and filter the noise, thereby improving her fidelity. However, if Eve is careless in her selections and ignores packets with covert information, she provides Alice and Bob with an unmonitored communication channel.

Active Eavesdropping Confusion is only possible when there is an asymmetry in knowledge between Eve and the confuser. To inject uncertainty in Eve's transcripts, the confuser exploits his knowledge of the noise filter. If Eve can also acquire this knowledge, then she can apply the same filter and can therefore trivially defeat confusion.

The intuitive solution to constructing a confusion-resistant eavesdropper is to make Eve active. In addition to passively observing traffic, an *active eavesdropper* attempts to learn more about the network and the communicating parties by sending out probes. For example, an active eavesdropper can counter the TTL confusion technique described in Section 3.2.2 by counting the number of network hops between itself and Bob. By acquiring additional knowledge, Eve can improve her selectivity and overall reliability.

Unfortunately, active eavesdropping is not always sufficient to ensure high interception fidelity. First, the probes used by an active Eve can themselves be subjected to a form of confusion. As a counter-counter-countermeasure, a confuser can inject a number of fake responses to Eve's probes. Returning to the TTL confusion example, a confuser can transmit fake ICMP TTL-exceeded messages to frustrate Eve's ability to discern the true TTL cutoff. Second, if Eve actively transmits probes, she may reveal her presence to Alice, Bob, and/or the confuser. Since eavesdropping is usually meant to be clandestine, active eavesdropping may be inappropriate for many situations. Finally, active eavesdropping may be ineffective for many types of confusion. For example, in semantic confusion, the noise filter may reside within Bob's application. In such cases, it may be extremely difficult for Eve to learn Bob's exact configuration.

Improving Fidelity through Eavesdropper Placement The location of Eve in the network topology may affect her resilience to confusion. An intuitive approach is to position her in close proximity to Alice. The ability of distant third-party confusers to inject noise is thus diminished as Eve can better discern Alice's communications from those of a distant forger. Unfortunately, this strategy is ineffective when Alice functions as the confuser. Unless Eve can determine which of Alice's messages are authentic, her position does little to improve her reliability.

A better solution is to place Eve as close as possible to Bob (and henceforth as far as possible from any confusers). For example, the TTL confusion technique will be ineffective if the noise filter (e.g., the network) is positioned before of Eve. A disadvantage of this approach is that Eve can only make reliable claims about the messages received by Bob. Her distance from Alice may make the authenticity of intercepted messages harder to establish.

A more ideal strategy is to deploy a number of collaborating eavesdroppers throughout the network. By comparing messages intercepted near the sender versus the receiver, Eve may be able to remove likely noise and improve her reliability. Although this technique would not be useful for semantic confusion (since the noise filter is located outside of the network), many lower-level confusion techniques may become ineffective. We leave the analysis of colluding eavesdropping as a future research direction.

6 Conclusion

Internet eavesdropping systems suffer from the eavesdropper's dilemma. For electronic wiretapping systems to be reliable, they must exhibit correct behavior with regard to both sensitivity and selectivity. Since capturing traffic is a requisite of any monitoring system, considerable research has focused on preventing evasion attacks and otherwise improving sensitivity. However, little attention has been paid to enhancing selectivity or even recognizing the issue in the Internet context. Traditional wisdom has held that eavesdropping is sufficiently reliable as long as the communicating parties do not participate in a bilateral effort to conceal their messages. We have demonstrated that even in the absence of cooperation between the communicating endpoints, reliable Internet eavesdropping is more difficult than simply capturing packets. If an eavesdropper cannot definitively and correctly select the pertinent messages from the captured traffic, the validity of the reconstructed conversation can be called into question. By injecting noise into the communication channel, unilateral or third-party confusion can make the selectivity process much more difficult and therefore further diminishes the reliability of electronic eavesdropping.

Whether eavesdropping can be performed reliably and confusion correctly detected and rejected on the Internet depends heavily on the specific interception topology and on the locations of potential sources of confusion traffic. Under some configurations, especially those near potential confusion sources, reliable eavesdropping is impossible, and confusion may even be impossible to detect. Under others, the eavesdropper must take into account the specific environment under which the interception was performed and must record lower layer traffic than may be possible under common practice with existing software. Even in those configurations where confusion can theoretically be filtered out, the eavesdropping software itself may still be susceptible to confusion, and, in fact, current software appears to be especially vulnerable to even the simplest confusion techniques. Through experiments testing current eavesdropping tools, we demonstrate three things: unilateral and third-party confusion and evasion countermeasures are practical; some configurations are inherently vulnerable to these countermeasures no matter how good the implementation; and no current implementation we tested is as good as it could be.

References

[AKCS00] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos. An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–167. ACM Press, 2000.

[BB00] M. Blaze and S. M. Bellovin. Inside RISKS: Tapping, tapping on my network door. *Communications of the ACM*, 43(10), December 2000.

[BBB+90] Charles H. Bennett, François Bessette, Gilles Brassard, Louis Salvail, and John A. Smolin. Experimental quantum cryptography. In *Advances in Cryptology - EUROCRYPT*, May 1990.

[Bel89] S. M. Bellovin. Security problems in the TCP/IP protocol suite. *Computer Communications Review*, 19:2:32–48, 1989.

[Bel00] S. M. Bellovin. Wiretapping the net. *The Bridge*, 20(2):21–26, 2000.

[Cas04] E. Casey. *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*. 2004.

[CM01] X. Carreras and L. Márquez. Boosting trees for anti-spam email filtering. In *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG, 2001.

[Com02] Computer Crime and Intellectual Property Section. Criminal Division. United States Department of Justice. *Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations*, July 2002. `http://www.cybercrime.gov/s&smanual2002.htm`.

[CSB05] E. Cronin, M. Sherr, and M. Blaze. Listen too closely and you may be confused. In *Proc. of 13th International Security Protocols Workshop* (to appear), 2005.

[Dic59] C. Dickens. *A Tale of Two Cities*. Apr 1859.

[DMS04] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proc. of the 13th Usenix Security Symposium*, pages 303–320, Aug 2004.

[ecs02] Electronic Crime Scene Investigation: A Guide for First Responders, Jul 2002. `http://www.ojp.usdoj.gov/nij/pubs-sum/187736.htm`.

[HKP01] M. Handley, C. Kreibich, and V. Paxson. Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In *Proc. of the 10th Usenix Security Symposium*, Aug. 2001.

[Int88] International Telecommunication Union. Multifrequency push-button signal reception. Recommendation Q.24, Telecommunication Standardization Sector of ITU, 1988.

[lib] The libnet packet construction library. `http://www.packetfactory.net/libnet/`.

[OV03] C. O'Brien and C. Vogel. Spam filters: Bayes vs. chi-squared; letters vs. words. In *ISICT '03: Proceedings of the 1st International Symposium on Information and Communication Technologies*, pages 291–296. Trinity College Dublin, 2003.

[Pax99] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(23–24):2435–2463, 1999.

[PN98] T. Ptacek and T. Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, Secure Networks, Inc., 1998.

[Pos81] J. B. Postel. Internet protocol. RFC 791, Internet Engineering Task Force, September 1981.

[PP03] R. Pang and V. Paxson. A high-level programming environment for packet trace anonymization and transformation. In *Proc. ACM SIGCOMM 2003*, Aug. 2003.

[Riv98] R. Rivest. Chaffing and winnowing: Confidentiality without encryption. `http://theory.lcs.mit.edu/~rivest/chaffing.txt`, March 1998.

[RR98] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. In *ACM Transactions on Information and System Security*, 1998.

[SAK05] Jeremy Stribling, Daniel Aguayo, and Maxwell Krohn. Rooter: A methodology for the typical unification of access points and redundancy, 2005. Originally accepted to the 9th World Multi-Conference on Systemics, Cybernetics and Informatics. Subsequently expelled.

[sci] SCIgen - An Automatic CS Paper Generator. `http://pdos.csail.mit.edu/scigen/`.

[SDHH98] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.

[Sim83] G.J. Simmons. The prisoners' problem and the subliminal channel. In *Proc. of IEEE Workshop on Communications Security CRYPTO'83*, 1983.

[Son] Dug Song. libdnet. `http://libdnet.sourceforge.net/`.

[Son99] D. Song. fragroute, 1999. `http://monkey.org/~dugsong/fragroute/`.

[SP03] U. Shankar and V. Paxson. Active mapping: Resisting NIDS evasion without altering traffic. In *Proc. of the 2003 IEEE Symposium on Security and Privacy*, May 2003.

[Tim02] K. Timm. IDS evasion techniques and tactics. *SecurityFocus Infocus*, May 2002. `http://www.securityfocus.com/infocus/1577`.

[Zaw03] J. Zawinski. DadaDodo: Exterminate all rational thought, 2003. `http://www.jwz.org/dadadodo/`.

[ZZY04] L. Zhang, J. Zhu, and T. Yao. An evaluation of statistical spam filtering techniques. *ACM Transactions on Asian Language Information Processing (TALIP)*, 3(4):243–269, 2004.

Appendix A: Legal and Policy Implications of Confusion and Interception Reliability

The question of interception reliability has implications in law and policy. We have largely avoided such issues in this paper, since they are outside our focus here. In this Appendix, we briefly survey a number of areas of law and policy in which determining the integrity and accuracy of Internet eavesdropping plays some role. This discussion is in no way intended to be comprehensive or authoritative, and we especially note our explicitly U.S.-centric references.

Wiretap evidence

The rules covering the treatment of electronic evidence in U.S. law are at best incomplete and, indeed, surprisingly inconsistent. We could find no decisive, broadly controlling cases that rule directly on how intercepted Internet traffic is to be treated when offered as legal evidence. The rules appear to largely depend on the context in which the evidence is presented. An excellent reference, particularly with regard to U.S. Federal criminal cases, is [Com02].

In general, evidence must be "authenticated" to be admitted as evidence; it must be shown to actually be what it is purported to be. At first blush, the possibility of confusion might appear to make this a difficult burden for evidence

derived from many Internet interception systems. However, Federal courts generally allow computer records evidence to be admitted unless there is *specific evidence* that it has been tampered with. In *U.S. v. Bonallo*, 858 F.2d 1427,1436 (9th Cir. 1988), the court ruled that "The fact that it is possible to alter data contained in a computer is plainly insufficient to establish untrustworthiness.". This was reiterated in *U.S. v. Allen*, 106 F.3d 965,700 (6th Cir. 1997), where the court ruled that "Merely raising the possibility of tampering is insufficient to render evidence inadmissible."

These authentication admissibility rules do not reconcile well with two basic properties of confusion-based interception countermeasures: the fact that, by its nature, confusion traffic is indistinguishable from real traffic, and the fact that it often can be injected by a third party that is neither one of the communicants nor the eavesdropper. That is, once the interception has been performed, there is no way to examine it to determine whether the interpretation is correct or the result of confusion. It would appear to be virtually impossible to meet the requirement for specific evidence of tampering, even when an interception actually has been tainted by confusion.

Even if wiretap evidence is admitted, it might still be attacked as to its "weight;" the finder-of-fact (the jury or, in some cases, the judge) would be allowed to hear opposing testimony intended to show how easily the intercept could have been fooled. Here, the possibility of confusion could have a significant impact on the believability of transcripts of certain Internet-based wiretaps.

Another issue related to authenticity is "authorship"; a party might deny that he or she really sent the data reflected by an intercept. Here, the courts have taken a far more skeptical view of Internet-based evidence, largely recognizing the lack of intrinsic authentication in data taken from the network. However, most of the cases concern stored data on networked servers, not network traffic itself. Here, corroborating circumstantial evidence is usually required to establish authorship. For example, in *U.S. v. Jackson*, 208 F.3d 633,638 7th Cir. 2000, the court would not allow the admission of web postings without additional evidence as to their author. Similarly, in *St. Clair v. Johnny's Oyster and Shrimp, Inc*, 76 F. Supp. 2d 773, 774, 775 (S.D. Texas 1999), the trial court found information from the Internet to be "inherently untrustworthy." (The judge's ruling in this case was remarkably unrestrained in its scathing criticism of the Internet, and we can recommend it as much for its amusement value as its legal insight). The possibility of confusion in the collection system would only strengthen this line of legal reasoning.

Minimization and confusion countermeasures

One counter-measure to certain kinds of confusion (e.g., TTL based, etc.) is to collect all traffic on the network and retrospectively analyze it, testing various hypotheses about the state of the network to expose the real traffic. Depending on the nature of the traffic collected, however, this approach may be contrary to U.S. law covering law enforcement interception of communication traffic.

In particular, an important requirement of the Federal wiretap statute ("Title III") is *minimization.* That is, when a Title III wiretap order is issued, the law enforcement agency is generally required to immediately discard any traffic not associated with the target of the order. This may make collecting enough contextual data to do accurate retrospective analysis against confusion legally problematic.

Design mandates to facilitate Internet wiretapping

Many law enforcement agencies have complained of the difficulty of capturing Internet traffic, and there have been recent proposals to apply the Communications Assistance to Law Enforcement Act (CALEA), which requires telephone companies to provide mandated wiretapping facilities in their networks, to the Internet. (ISPs are now largely exempt from the CALEA requirements except with respect to voice-over-IP traffic.)

The heterogeneous nature of the Internet architecture makes guaranteeing wiretap access to law enforcement problematic to begin with, although many ISPs are able to provide duplicated network streams to comply with certain kinds of wiretap requests. Constructing a wiretap interface that is immune from confusion countermeasures, however, may be much more problematic. A detailed analysis is beyond the scope of this paper, but at a minimum we suggest that any proposed wiretapping design mandates for the Internet make explicit how confusion is expected to be treated.

Appendix B: Tested Eavesdropping Systems

In Section 4.1, we evaluated confusion techniques against a number of common eavesdropping tools. In this appendix we briefly discuss the tools used and configurations when important.

Open Source Eavesdropping Tools

- **Bro:** Bro is a network intrusion detection system developed at the University of California, Berkeley. As such, it does not operate as an eavesdropping tool by default. However, it has a very robust stream reconstruction engine, and can be cajoled into acting as an offline analysis tool. We ran Bro using the 'weird', 'conn', 'contents', 'frag', and 'smtp' policies using their default settings. Bro can be found at `http://www.bro-ids.org`.
- **Chaosreader:** Chaosreader is a user-friendly TCP reconstruction tool which creates HTML pages for the contents of intercepted sessions. It can be found at `http://chaosreader.sourceforge.net`.
- **Ethereal:** Ethereal is a very popular eavesdropping tool. Although most of its features are packet oriented, it contains a TCP reassembly option which was used for the experiments. Ethereal can be found at `http://www.ethereal.com/`.
- **Snort:** Snort is another commonly used NIDS. We ran it in offline mode using the stream4 and stream4_reassemble preprocessors with the log_flushed_streams option. In addition, we used the snort-replay patch, which uses its own stream reconstruction implementation. Snort can be found at `http://www.snort.org/`, and snort-replay at `http://www.algonet.se/~nitzer/snort-replay/`.
- **tcpick:** tcpick is a pcap-based packet sniffer and tcp reconstruction tool. It can be found at `http://tcpick.sourceforge.net/`.
- **tcptrace:** tcptrace is an analysis tool for pcap-based network intercepts. Among its many features, tcptrace can reconstruct captured TCP streams. It can be found at `http://jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html`.
- **tcpflow:** tcpflow is a useful tool for conducting TCP stream reassembly. It operates by processing pcap dump files and extracting the contents of TCP streams. It can be found at `http://www.circlemud.org/~jelson/software/tcpflow/`.

Commercial Eavesdropping Tools

- **CommView:** CommView is a commercial Windows eavesdropping tool. An evaluation version can be found at `http://www.tamos.com/products/commview/`.
- **NetworkActiv PIAFCTM:** PIAFCTM is a commercial Windows eavesdropping tool. A trial version is available at `http://www.networkactiv.com/PIAFCTM.html`.
- **Sniffem:** Sniffem is a commercial Windows eavesdropping tool. A trial version is available at `http://www.sniff-em.com/sniffem.shtml`.