On the (un)Reliability of Eavesdropping

Eric Cronin*, Micah Sherr, and Matt Blaze
Department of Computer and Information Science
University of Pennsylvania
E-mail: {ecronin,msherr,blaze}@cis.upenn.edu
*Corresponding author

Abstract: We investigate the reliability of current generation eavesdropping tools and show that obtaining "high fidelity" transcripts is harder than previously assumed. Even in situations highly favorable to the eavesdropper, simple unilateral countermeasures are shown to be sufficient to prevent all tested systems from reliably reconstructing communicated messages. Less than a third of the tested systems report irregularities, and 45% incorrectly interpret covertext chosen by the sending party. Unlike cryptography or steganography, the techniques introduced require no cooperation by the communicating parties and, in some case, can be employed entirely by a third party not involved in the communication at all.

Keywords: eavesdropping, electronic interception, eavesdropping countermeasures, network anti-forensics

Reference to this paper should be made as follows: Cronin, E., Sherr, M. and Blaze, M. (2007) 'On the (un)Reliability of Eavesdropping', Int. J. Security and Networks, Vol. X, No. Y, pp. zz-zz

Biographical notes: Eric Cronin is a PhD candidate in Computer and Information Sciences at the University of Pennsylvania. His research interests include network security, privacy, and distributed systems. He is a member of ACM, IEEE, and USENIX.
Micah Sherr is a PhD candidate in Computer and Information Sciences at the University of Pennsylvania. His research interests include network security, protocol design and analysis, network intrusion detection and prevention, and privacy and data confidentiality. He is a member of USENIX.
Matt Blaze is an associate professor of Computer and Information Sciences and director of the Trusted Network Eavedropping and Countermeasures project at the University of Pennsylvania. His research interests include secure systems, cryptology and cryptographic protocols, and large-scale systems. He is a member of ACM, IACR and IEEE, and is a director of the USENIX association.

1 INTRODUCTION

The results of Internet interceptions are almost always accepted uncritically. While previous work has shown the potential for spurious errors (Bellovin, 2000; Blaze and Bellovin, 2000) or evasion (Ptacek and Newsham, 1998; Paxson, 1999) to interfere with capture, there has been remarkably little exploration of the problems which face an eavesdropper who wishes to ensure the accuracy of their intercepts. We assert that the task of the eavesdropper is actually far more difficult than has previously been realized, and show that existing tools are insufficient to gauge the accuracy of captured traffic.

At least six properties of the Internet protocol stack and architecture make it difficult for an eavesdropper to accurately reconstruct communications from its intercepts:

- decentralized control and heterogeneous implementations,
- "best effort" (as opposed to reliable) message delivery that allows data to be re-ordered, duplicated or dropped in transit,
- shared state and context between communicating parties,
- dynamic (and often asymmetric) routing that can change during a flow's lifetime,
- lack of sender and receiver authentication, and
- ambiguities in protocols, implementations, and configurations.

These properties mean that a great deal of state information is involved in the correct interpretation of any given packet, and this state is spread across many places, including each of the communicating parties and the network itself. Without complete knowledge of this state, the mere presence of a packet somewhere on the network does not automatically imply that it will be accepted by the recipient given in its header, that it came from the supposed sender, or that it has not been (or will not be) altered, duplicated, or deleted somewhere along its path.

Any intercept system must take into account these properties (and all the corresponding state) in order to ensure not only that it is sufficiently *sensitive* (that it receives all data exchanged between the targets), but that it is also sufficiently *selective* (that it rejects spurious data that is not actually part of the targets' exchange). The figure of merit most often considered in judging intercept systems is sensitivity; adequate selectivity, on the other hand, is generally thought to be easily achieved by cursory examination of, e.g., packet headers. In fact, selectivity may be a far more difficult problem than most intercept systems recognize, especially in the presence of deliberate countermeasures.

Fortunately for the eavesdropper, on more benign networks at least, many of the factors that might introduce uncertainty about the veracity and interpretation of a given packet are relatively static, at least for the lifetime of a particular interception. For example, although routes can theoretically change midstream, in practice, they rarely do, and although routers and hosts are free to alter, reorder, delay, and duplicate packets, for the most part they refrain from doing so.

However, this lends a false sense of security to those producing eavesdropping tools. Depending on the network configuration, many ambiguities can be intentionally induced, either by one of the communicating parties or by a third party altogether. In fact, across much of the protocol stack, from the physical layer to the applications, it is surprisingly simple to introduce data that appears entirely valid but that might not be received and processed by the purported recipient. The Internet appears almost to have been designed to maximize uncertainty from the point of view of those eavesdropping on it.

In particular, we observe that a single party, which we call a *confuser*, can introduce traffic directed at an eavesdropper but that is never actually received (or if received, is rejected) by the ostensible recipient. Depending on the eavesdropper's configuration and its position in the network, this traffic can be made indistinguishable from legitimate traffic. In the presence of sufficient confusion, an eavesdropper may be able to be made arbitrarily uncertain as to whether a given intercepted message was real or spurious.

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 *confusion* if it captures and records in its transcripts messages *purportedly* from Alice to Bob but that are rejected or otherwise not processed by Bob.

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.

- 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.[1] 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 the confuser is a third-party, then neither Alice nor Bob needs to be aware of the confusion. Unlike bilateral techniques 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, a property sometimes referred to in the literature as *plausible deniability* (Roe, 1997).

2 RELATED WORK

There has been little prior work investigating the general problem of traffic interception from the eavesdropper's point of view (Bellovin, 2000; Blaze and Bellovin, 2000; Cronin et al., 2005). However, considerable research has addressed the related (but not identical) topic of information privacy. Cryptography, steganography, subliminal or covert channels (Simmons, 1983), winnowing and chaffing (Rivest, 1998), quantum communication (Bennett et al., 1990), and anonymous communications (Dingledine et al., 2004; Reiter and Rubin, 1998), for example, all focus on establishing confidential communication.

Work from the eavesdropper's point of view has primarily been limited to the specialized subtopic of intrusion detection (Shankar and Paxson, 2003; Pang and Paxson, 2003; Paxson, 1999). 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

[1] While work in anonymity (Dingledine et al., 2004; Reiter and Rubin, 1998; Pfitzmann et al., 1991) has yielded useful tools for concealing the endpoints of communication, these systems typically rely on networks of participating nodes to achieve anonymity. In contrast, confusion is designed to operate with existing software and network systems.

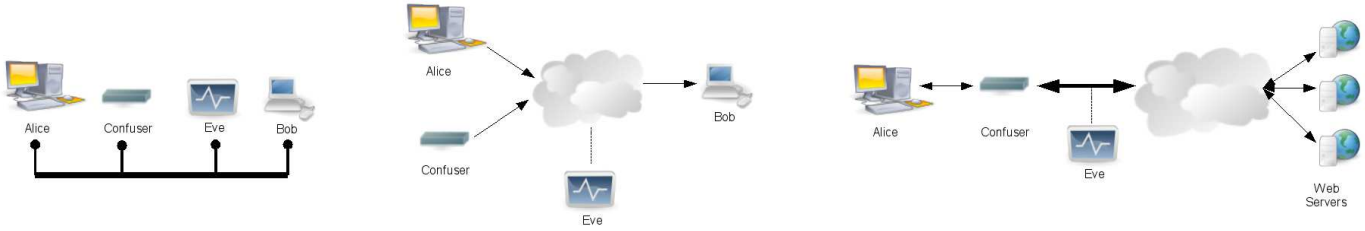

Figure 1: *Left:* An eavesdropping topology in which all parties communicate via the same shared bus. *Center:* A configuration in which Eve is located on the network between Alice and Bob. *Right:* A topology in which the confuser also functions as a router.

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.

The confusion techniques presented in this paper can be viewed as a form of network anti-forensics. There have been a number of studies on tools which are targeted both at thwarting storage forensics in general and with both thwarting and deceiving particular forensic tools (Geiger and Cranor, 2005), and standard procedures for forensic analysis have been updated to detect such techniques (Technical Working Group for Electronic Crime Scene Investigation, 2001; Technical Working Group for the Examination of Digital Evidence, 2004). As this work shows, similar attention must be paid to suspicious network evidence.

Finally, we note that concern about eavesdropping interacts with the legal system in several ways. The Appendix 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 (Technical Working Group for Electronic Crime Scene Investigation, 2001; Casey, 2004).

3 CONFUSION IN THE INTERNET

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. (Postel, 1981a)") 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.

3.1 Physical Layer Confusion

At the physical layer, network devices convert analog signals into digital encodings. To allow interoperable devices, standards exist that define acceptable ranges for amplitudes, frequencies, voltages, and so forth (LAN/MAN Standards Committee, 2005, 1990, 2003). However, because transmission and decoding are analog processes, for any given parameter (frequency, amplitude, etc.), no two decoders will use precisely the same threshold to determine whether a given signal is accepted or rejected. Thus, network devices, particularly commodity hardware, do not strictly abide by these standards and often interpret messages sent outside of the specified ranges.

Alice can exploit these differences to evade as well as confuse Eve. 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). To *evade* Eve, Alice can transmit messages at a frequency, amplitude, or voltage that is imperceptible 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.

Eve's obvious counter-countermeasure (i.e., enhancing her sensitivity) has the unfortunate effect of increasing her vulnerability to confusion. If Eve is *more* sensitive than Bob, evasion is not possible. However, a third-party confuser can now 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 legitimate messages.

3.2 Link Layer Confusion

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.) 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 (a similar approach was used by (Ptacek and Newsham, 1998) in the context of network intrusion detection systems). 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 be reliable unless it both intercepts and processes link layer headers.

3.3 Network Layer Confusion

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 (Postel, 1981a; Ptacek and Newsham, 1998). 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.

3.4 Transport Layer Confusion

TCP ensures reliable communication by resolving a number of transmission and transport errors, including damaged, lost, duplicated, and out-of-order packets. While the TCP standard details compensatory measures (e.g., packet retransmission) for benign and transient network errors, it does not specify how to address certain ambiguities that may arise due to an antagonist.

For example, the TCP specification states that TCP sequence numbers should be used to detect and eliminate duplicates (Postel, 1981b). Redundant packets that have sequence numbers corresponding to data that have already been received are immediately discarded. An underlying assumption of the specification is therefore that duplicate packets have identical payloads. If this is not the case, it is unclear which "duplicate" should be considered legitimate. We present two potential ways in which the network can delay or drop packets after the eavesdropper has seen them. In both approaches, Alice functions as the confuser as well as the sender.

Packets can become reordered due to the Quality of Service (QoS) IP field. The optional QoS field allows packets to be given priorities that routers use in determining when they should be serviced. By sending the noise packets first but with a low priority, and the legitimate packets later with higher priorities, it is possible for them to become reordered in the network. Unfortunately, the amount of control end-users have over the QoS settings for their packets can be limited. Often QoS is normalized at ingress when clients are charged for their usage.

Packet reordering may also be accomplished by exploiting *fast-path* routing. Modern routers have been heavily optimized towards forwarding on normal traffic. Packets that do not contain any unusual flags or options take the hardware-based "fast-path" through the router; packets requiring abnormal examination by the router take the "slow-path" through software. In many routing architectures, all packets containing IP options are processed via the slow-path (Rossi and Welzl, 2003, 2004), which can be used to delay noise packets. Thus, Alice can thwart eavesdropping by injecting noise with non-NULL IP options. If the noise is transmitted before each legitimate TCP packet, Eve may interpret the noise rather than the true message stream. Because Bob is located after one or more routers that employ fast-path routing, the legitimate messages will arrive before the noise, hence Bob's reconstruction of the stream will be correct.

3.5 Confuser-in-the-middle

A more fundamental limitation of reliable electronic eavesdropping occurs when a confuser can position itself between Alice and Eve (see Figure 1, *right*). In this *confuser-in-the-middle* approach, Alice requests information from one or more services (e.g., web servers).

In our topology, Alice and the confuser play multiple roles. Alice functions as both the sender and the receiver.

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`.

Figure 2: Eavesdropping software evaluated

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.

4 FAILURE OF EAVESDROPPING SYSTEMS

In this section we examine common tools for eavesdropping in several domains, and show how they fall vulnerable to simple, unilateral attacks.

4.1 Inadequate selectivity in current generation eavesdropping systems

To demonstrate the susceptibility of current eavesdropping tools to confusion, we implemented the MAC and TTL confusion techniques described in the previous section. (Fragroute (Song, 2002) also provides an implementation of the NIDS techniques, but it was found to be unsuitable for general purpose bidirectional 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

Software	No Confusion (byte-sized pkts)		MAC Confusion		TTL Confusion	
	Inter-pretation	**Detected Anomalies**	**Inter-pretation**	**Detected Anomalies**	**Inter-pretation**	**Detected Anomalies**
bro	Success	None reported	Failure (Covertext)	Retrans. inconsistency	Failure (Covertext)	Retrans. 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
Network-Activ 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 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 tool incorrectly interpreted the covertext as the legitimate messagetext. See Figure 4.
Failure (Random noise) - No discernible English text could be obtained from the eavesdropper's interpretation.

Figure 3: Ineffectiveness of various eavesdropping software against confusion techniques.

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 (recall that since TCP is stream based, applications that rely on TCP are generally unaffected by the size of the transmitted packets). For every byte of legitimate text, Alice sends eight noise packets. Of the eight noise streams, the first is comprised of a "cover message". This first stream, although composed of noise, constitutes a false but sensible message (a passage from Dickens' "A Tale of Two Cities" (Dickens, 1994)). The remaining seven 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 seven random noise streams. No modifications were made to the SMTP server (Bob).

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 Figure 2). 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 five 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.

The performance of the eavesdroppers in the presence of confusion was startlingly lacking. Figure 3 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 of the two confusion techniques was applied. Anomalies were reported only by 18% of the eavesdroppers with the MAC-based approach and 27% of the systems 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 4). In all cases, the email server (Bob) correctly received Alice's communication and delivered the email to its intended recipient.

4.2 Injecting Confusion in 802.11 Networks

With wired Ethernet, the widespread deployment of switches to replace hubs and older shared bus technologies has somewhat reduced the risk of malicious users pas-

sively 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 *Confusion Access Point* (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 *confusion daemon* 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.

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 (Schiffman, 2005) and libdnet (Song, 2005). 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 legitimate traffic and noise indistinguishable at 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 5. 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.

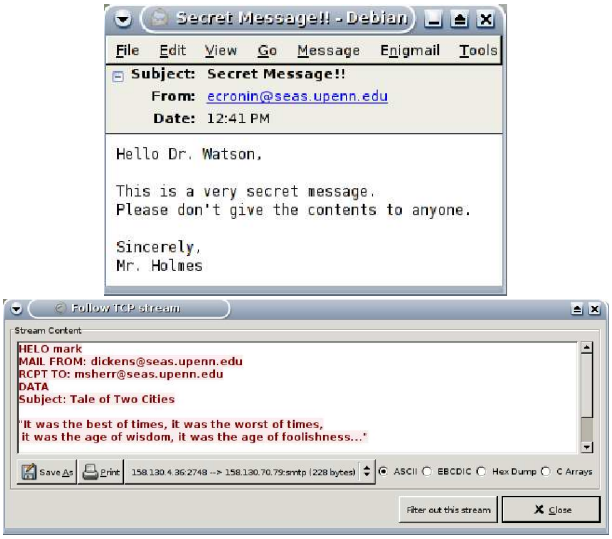

Figure 4: Top: Legitimate message received by the SMTP server (Bob) and the intended email recipient. Bottom: An eavesdropping system's (Ethereal) reconstruction in which Eve not only fails to capture Alice's message, she also perceives the covertext as the legitimate message.

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

Figure 5: Actual number of connections and number of connections observed by Eve for Confusing Access Point.

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.

4.3 POTS Evasion and Confusion

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 (see Section 3.1). For a simple example, we consider voice telephone signaling be-

tween 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 (International Telecommunication Union, 1988), 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. Telephone wiretapping countermeasures are studied in detail in (Sherr et al., 2005).

5 IMPROVING EAVESDROPPING RELIABILITY

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

5.1 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, must 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. In such environments the steps Eve can take to improve sensitivity are reduced.

5.2 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 can be defeated with improved software. By enhancing her sensitivity, Eve may be able to better identify and filter the noise, thereby improving her reliability. However, if Eve is careless in her selections and ignores packets with covert information, she provides Alice and Bob with an unmonitored communication channel.

5.3 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 (e.g., the network topology or Bob's TCP/IP stack configuration) to ensure that the noise will be removed or filtered before being processed by Bob. 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 3.3 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 reliable reconstruction of the intercepted traffic. 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.

5.4 Improving Reliability 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 Bob and Eve reside on the same local network. 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. We leave the analysis of colluding eavesdropping as a future research direction.

6 CONCLUSION

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, diminishing 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. 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.

ACKNOWLEDGMENTS

The authors would like to thank Harry Hoffman for his assistance configuring Bro and Snort for the experiments in Section 4.1. This work was partially supported by the US National Science Foundation Cyber-Trust program under contract NSF-0524047.

REFERENCES

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

Bennett, C. H., Bessette, F., Brassard, G., Salvail, L., and Smolin, J. (1990). Experimental quantum cryptography. In Damgard, I. B., editor, *Advances in Cryptology - EUROCRYPT '90: Workshop on the Theory and Application of Cryptographic Techniques*, volume 473 of *Lecture Notes in Computer Science*, pages 253–265. Springer Berlin / Heidelberg.

Blaze, M. and Bellovin, S. M. (2000). Inside RISKS: Tapping on my network door. *Communications of the ACM*, 43(10):136.

Casey, E. (2004). *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*. Elsevier Academic Press, London; San Diego, CA, second edition.

Computer Crime and Intellectual Property Section (2002). *Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations*. Criminal Division, United States Department of Jus-

tice. *(available at* `http://www.cybercrime.gov/s&smanual2002.htm`*).*

Cronin, E., Sherr, M., and Blaze, M. (2005). Listen too closely and you may be confused. In *Security Protocols 2005, 13th International Workshop. Revised Selected Papers*, Lecture Notes in Computer Science. (*to appear*).

Dickens, C. (1994). *A Tale of Two Cities*. Project Gutenberg, Salt Lake City, UT.

Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation onion router. In *Proc. of the 13th USENIX Security Symposium*, pages 303–320. The USENIX Association.

Geiger, M. and Cranor, L. F. (2005). Counter-forensic privacy tools: A forensic evaluation. Technical Report CMU-ISRI-05-119, Institute for Software Research, International, Carnegie Mellon University, Pittsburgh, PA.

International Telecommunication Union (1988). Multifrequency push-button signal reception. Recommendation Q.24, Telecommunication Standardization Sector of ITU.

LAN/MAN Standards Committee (1990). Token-passing bus access method and physical layer specifications. IEEE Std. 802.4, IEEE Computer Society.

LAN/MAN Standards Committee (2003). Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std. 802.11, IEEE Computer Society.

LAN/MAN Standards Committee (2005). Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE Std. 802.3, IEEE Computer Society.

Pang, R. and Paxson, V. (2003). A high-level programming environment for packet trace anonymization and transformation. In *Proc. ACM SIGCOMM 2003*, pages 339–351. ACM Press.

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

Pfitzmann, A., Pfitzmann, B., and Waidner, M. (1991). ISDN-mixes: Untraceable communication with very small bandwidth overhead. In *GI/ITG Conference on Communication in Distributed Systems*, pages 451–463, Mannheim, Germany. Springer-Verlag, Heildelberg.

Postel, J. B. (1981a). Internet protocol. RFC 791, Internet Engineering Task Force.

Postel, J. B. (1981b). Transmission Control Protocol. RFC 793, Internet Engineering Task Force.

Ptacek, T. and Newsham, T. (1998). Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, Secure Networks, Inc., Calgary, Alberta, Canada.

Reiter, M. K. and Rubin, A. D. (1998). Crowds: Anonymity for web transactions. In Allen, R., editor, *Transactions on Information and System Security*, volume 1(1), pages 66–92. ACM Press.

Rivest, R. (1998). Chaffing and winnowing: Confidentiality without encryption. `http://theory.lcs.mit.edu/~rivest/chaffing.txt` (accessed 15 Feb, 2007).

Roe, M. (1997). *Cryptography and Evidence*. PhD thesis, Computer Laboratory, University of Cambridge, Cambridge, UK.

Rossi, M. and Welzl, M. (2003). On the impact of IP option processing. Technical Report 15, Leopold-Franzens Universit at Innsbruck, Infmath Imaging.

Rossi, M. and Welzl, M. (2004). On the impact of IP option processing - part 2. Technical Report 26, Leopold-Franzens Universit at Innsbruck, Infmath Imaging.

Schiffman, M. D. (2005). The libnet packet construction library. `http://www.packetfactory.net/projects/libnet/` (accessed 15 Feb, 2007).

Shankar, U. and Paxson, V. (2003). Active mapping: resisting NIDS evasion without altering traffic. In *Proc. of the 2003 IEEE Symposium on Security and Privacy*, pages 44–61. IEEE Computer Society.

Sherr, M., Cronin, E., Clark, S., and Blaze, M. (2005). Signaling vulnerabilities in wiretapping systems. *IEEE Security and Privacy*, 3(6):24–36.

Simmons, G. J. (1983). The prisoners' problem and the subliminal channel. In Chaum, D., editor, *Advances in Cryptology - CRYPTO '83*, pages 51–67, New York. Plenum Press.

Song, D. (2002). Fragroute: Intercept, modify, and rewrite egress traffic. `http://monkey.org/~dugsong/fragroute/` (accessed 15 Feb, 2007).

Song, D. (2005). Dumb networking library. `http://libdnet.sourceforge.net/` (accessed 15 Feb, 2007).

Technical Working Group for Electronic Crime Scene Investigation (2001). Electronic Crime Scene Investigation: A Guide for First Responders. National Institute of Justice Guide NCJ 187736, U.S. Department of Justice Office of Justice Programs.

Technical Working Group for the Examination of Digital Evidence (2004). Forensic Examination of Digital Evidence: A Guide for Law Enforcement. National Institute of Justice Special Report NCJ 199408, U.S. Department of Justice Office of Justice Programs.

APPENDIX: LEGAL AND POLICY IMPLICATIONS

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 (Computer Crime and Intellectual Property Section, 2002).

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 sug-

gest that any proposed wiretapping design mandates for the Internet make explicit how confusion is expected to be treated.