Chapter 16

ON THE RELIABILITY OF NETWORK EAVESDROPPING TOOLS

Eric Cronin, Micah Sherr and Matthew Blaze

Abstract This paper analyzes the problem of intercepting Internet traffic from the eavesdropper's point of view. It examines the reliability and accuracy of transcripts, and shows that obtaining "high fidelity" transcripts is harder than previously assumed. Even in highly favorable situations, such as capturing unencrypted traffic using standard protocols, simple – and entirely unilateral – countermeasures are shown to be sufficient to prevent accurate traffic analysis in many Internet interception configurations. In particular, these countermeasures were successful against every available eavesdropping system we tested. Central to our approach is a new class of "confusion" techniques, that unlike cryptography or steganography, do not require cooperation by the communicating parties and, in some cases, can be employed entirely by a third party who is not involved in the communication.

Keywords: Eavesdropping, electronic interception, countermeasures

1. Introduction

The results of Internet interceptions are almost always accepted uncritically. While previous work has shown the potential for spurious errors [1, 3] or evasion [13, 15] to interfere with capture, there has been remarkably little exploration of the problems that face eavesdroppers who wish to ensure the accuracy of their intercepts. We assert that the eavesdropping task is 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 intercepts. They include 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 states) 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) [4]. 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, for example, 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, 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. In fact, across much of the protocol stack, from the physical layer to 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 position in the network, this traffic can be made indistinguishable from legitimate traffic. In the presence of sufficient confusion, an eavesdropper could be made arbitrarily uncertain as to whether a given intercepted message was real or spurious.

We introduce some terminology that will be used throughout 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 (in most protocols the roles are symmetric and often alternating). Eve will be the eavesdropper. An interception system is vulnerable to confusion if it captures and records in its transcripts messages that are 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 note that confusion has some interesting qualities that make it attractive as an eavesdropping countermeasure.

- While cryptography is typically used to ensure the confidentiality of message payloads, confusion protects both message contents and metadata. It may, therefore, be advantageous to combine confusion with encryption to mask the signaling information as well as the 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 need 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.

2. Related Work

There has been little prior work investigating the general problem of traffic interception from the eavesdropper's point of view [1, 3, 5]. However, considerable research has addressed the related topic of information privacy. Cryptography, steganography, subliminal or covert channels [20], winnowing and chaffing [17], quantum communication [2] and anonymous communications [7, 16], for example, all focus on establishing confidential communication.

Work from the eavesdropper's point of view has primarily been limited to the specialized area of intrusion detection [12, 13, 18]. 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 can flag suspicious traffic. However, unlike a NIDS, a general purpose eavesdropper must process all traffic, both normal and anomalous. It is possible to draw some results from NIDS research, but the applicability is limited by the different constraints on topology and communication characteristics.

3. Confusion in the Internet Architecture

The Internet is, by design, 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 an 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" [14]) 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 with 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, etc. [8–10]. However, because transmission and decoding are analog processes, for any given parameter (frequency, amplitude, etc.), no two decoders will use the same threshold to determine whether a given signal is accepted or rejected. Thus, network devices, especially commodity hardware, do

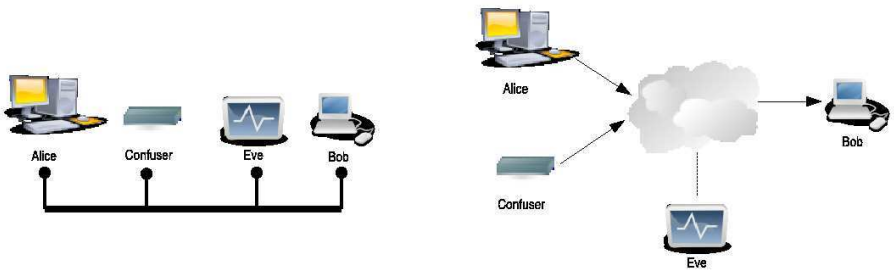

Figure 1. Left: All parties communicate using a shared bus; Right: Eve is located between Alice and Bob.

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 the left-hand side of Figure 1, 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 are 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 [4]. 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.

We show empirically in Section 4 that current eavesdropping systems suffer from inadequate selectivity. Although most eavesdropping systems can record traffic at the link layer, they often ignore Ethernet frames and instead process messages at 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 is not delivered to Bob [15].

Neither Bob nor the local gateway process the noise as their operating systems silently discard Ethernet frames whose MAC addresses do not match those of the network interface.

This technique is 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 intercepts and processes link layer headers.

3.3 Network Layer Confusion

If Eve intercepts a packet on the path from Alice and Bob (right-hand side of Figure 1), she must carefully examine the packet's IP header to form an opinion as to whether the packet is deliverable. A packet may not be delivered for several reasons: the packet checksum may be incorrect, IP options may be specified that are unsupported by an intermediary router (e.g., source routing), the packet size may exceed the MTU of a hop, or the initial time-to-live (TTL) value may be insufficient to reach Bob [14, 15]. 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 link layer techniques, 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, who does not examine IP headers or lacks enough selectivity to determine whether packets are deliverable, is inherently vulnerable to this type of confusion.

4. Failure of Current Eavesdropping Systems

In this section we examine common eavesdropping tools in several domains, and show how they are vulnerable to simple, unilateral attacks. We look at examples of digital and analog tools.

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 with their default settings. Bro can be found at: www.bro-ids.org.
- **Chaosreader:** Chaosreader is a user-friendly TCP reconstruction tool that creates HTML pages for the contents of intercepted sessions. It can be found at: chaosreader.sourceforge.net.
- **Ethereal:** Ethereal is a popular eavesdropping tool. Although most of its features are packet oriented, it contains a TCP reassembly option that was used in the experiments. Ethereal can be found at: www.ethereal.com.
- **Snort:** Snort is a 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: www.snort.org, and snort-replay: at www.algonet.se/∼nitzer/snort-replay.
- **tcpick:** tcpick is a pcap-based packet sniffer and TCP reconstruction tool. It can be found at: 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: 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: www.circlemud.org/∼jelson/software/tcpflow.

Commercial Eavesdropping Tools

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

Figure 2. Eavesdropping tools evaluated in this work.

4.1 Digital Eavesdropping Evasion

To demonstrate the susceptibility of current eavesdropping tools (Figure 2) to confusion, we implemented the MAC and TTL confusion techniques described in Section 3 and originally introduced as NIDS attacks in [15]. Note that fragroute [21] 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 current eavesdropping tools are susceptible to even weak forms of confusion.

In our experiments, Alice transmits an email via SMTP to our institution's email server (Bob). To confuse Eve, Alice (who functions as the confuser) injects spurious noise using either the MAC or the TTL confusion techniques. To maximize confusion, Alice sends the legitimate email and the noise in byte-sized packets (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 comprises 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* [6]. 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 the 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 (Figure 2). Experiments were conducted on a network test bed 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 that are connected via a 100MB/s switch.

The performance of the eavesdroppers in the presence of confusion was startlingly lacking. Table 1 describes Eve's (in)ability to reliably reconstruct email messages. Although all but one eavesdropping package could reconstruct Alice's message in the absence of confusion, all the tested systems failed to interpret her message when either of the two confusion techniques was applied. Anomalies were reported by only 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 (Figure 3). In all cases, the email server (Bob) correctly received Alice's communication and delivered the email to its intended recipient.

Table 1. Ineffectiveness of various eavesdropping tools against confusion techniques. ✓: Tool correctly interpreted the message; $\times_C$: Tool incorrectly interpreted cover message as legitimate message (see Figure 3); $\times_R$: No discernible English text obtained from the eavesdropper's interpretation; *DUP*: TCP DUPs detected; *TTL*: TTL exceeded; *RI*: Retransmission inconsistency.

Software	No Confusion		MAC Confusion		TTL Confusion	
	Valid	Errors	Valid	Errors	Valid	Errors
Bro	✓	—	$\times_C$	*RI*	$\times_C$	*RI*
chaosreader	✓	—	$\times_R$	—	$\times_R$	—
CommView	✓	—	$\times_C$	—	$\times_C$	—
Ethereal	✓	—	$\times_C$	—	$\times_C$	—
PIAFCTM	✓	—	$\times_C$	—	$\times_C$	—
Sniffem	$\times_R$	—	$\times_R$	—	$\times_R$	—
snort-replay	✓	—	$\times_R$	—	$\times_R$	—
snort-stream4	✓	—	$\times_R$	—	$\times_R$	*TTL*
tcpick	✓	—	$\times_C$	—	$\times_C$	—
tcptrace	✓	—	$\times_R$	*DUP*	$\times_R$	*DUP*
tcpflow	✓	—	$\times_R$	—	$\times_R$	—

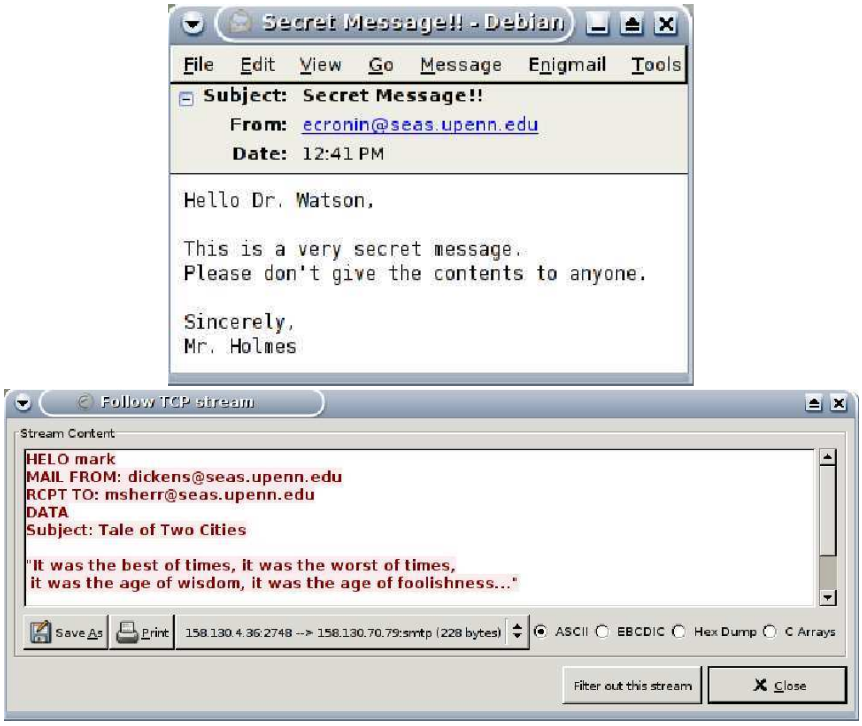

Figure 3. Top: Legitimate message received by the SMTP server (Bob) and the intended email recipient; Bottom: Ethereal reconstruction in which Eve fails to capture Alice's message and perceives the cover message as the legitimate message.

4.2 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. As an 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 based on the international DTMF standard [11]. Each of the 16 DTMF digits is composed 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 that satisfies the specifications exactly is surprisingly difficult and expensive. Instead, most decoders relax the tolerances in one or more of the dimensions of the standard, accepting tones that 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 evade and confuse eavesdroppers.

To test the practicality of this countermeasure, we constructed a simulated telephone network. A Teltone TLS-5 Telephone Line Simulator was used as the central office, providing dial tone 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 to it; 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 precise, configurable parameters. In our experiments, we focused on two variables: the frequency and amplitude of the higher of the two DTMF frequencies. Settings that allowed for both evasion and confusion were discovered using binary search. Evasion can be applied by decreasing the amplitude of the higher frequency. At -39 dBm, the central office correctly decoded Alice's signal and completed the call, while the eavesdropper recorded nothing. Similarly, when the high frequency was increased by 3 Hz, the central office no longer recognized Alice's touch-tones, but the eavesdropper recorded them as having been dialed. Using Alice's handset in

coordination with the AM8a, the legitimate number could be dialed and interspersed with out-of-range digits to provide confusion. In addition, although we did not test the scenario, by combining both techniques it is clear that Alice could drive Eve to a specific false phone number.

This experiment highlights the challenges faced by an eavesdropper positioned too close to the sender. Limited sensitivity and imperfect selectivity make it susceptible to evasion and confusion countermeasures. While Eve may be certain that the intercepts originated from Alice, she cannot be certain where they terminate in the telephone network. 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. See [19] for details about using confusion as a telephone wiretapping countermeasure.

5. Improving Eavesdropping Reliability

The experiments described in the previous section show how unilateral countermeasures can reduce the reliability of eavesdropping systems. This section explores methods to improve the resilience of eavesdropping tools 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 that is available (even data that appears to be erroneous). Any action that could have been performed automatically by the lower layers, such as discarding corrupt packets, can be carefully emulated by Eve in a more selective manner.

Unfortunately, this approach may be hard to implement. For example, many authorized uses of eavesdropping in the United States operate under strict limitations on what can be recorded to prevent the traffic of those who are not under suspicion from being observed. In such an environment, the steps that Eve can take to improve her 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 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 is removed or filtered before being processed by Bob. If Eve can also acquire this knowledge, 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 by counting the number of network hops to Bob. By acquiring additional knowledge, Eve can improve her selectivity and overall reliability.

Unfortunately, active eavesdropping does not always ensure the reliable reconstruction of 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 in many situations.

5.4 Improving Reliability via 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 as far as possible from any confusers). For example, the TTL confusion technique is 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. The analysis of colluding eavesdropping is an area of future research.

6. Conclusions

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 bilateral efforts 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 detected correctly and rejected on the Internet depend heavily on the specific interception topology and on the locations of the potential sources of confusion traffic. Even in those configurations where confusion can theoretically be filtered out, eavesdropping tools may be susceptible to confusion. In fact, current eavesdropping tools appear to be especially vulnerable to even the simplest confusion techniques.

Acknowledgements

The authors would like to thank Harry Hoffman for his assistance configuring Bro and Snort for the experiments in Section 4. This work was partially supported by the NSF Cyber Trust Program under Grant CNS-0524047.

References

[1] S. Bellovin, Wiretapping the net, *The Bridge*, vol. 20(2), pp. 21-26, 2000.

[2] C. Bennett, F. Bessette, G. Brassard, L. Salvail and J. Smolin, Experimental quantum cryptography, *Advances in Cryptology – Proceedings of EUROCRYPT'90*, Springer-Verlag, Berlin-Heidelberg, pp. 253-265, 1990.

[3] M. Blaze and S. Bellovin, Tapping on my network door, *Communications of the ACM*, vol. 43(10), p. 136, 2000.

[4] E. Cronin, M. Sherr and M. Blaze, The Eavesdropper's Dilemma, Technical Report MS-CIS-05-24, Department of Computer and Information Science, University of Pennsylvania, Philadephia, Pennsylvania, 2005.

[5] E. Cronin, M. Sherr and M. Blaze, Listen too closely and you may be confused, *Proceedings of the Thirteenth International Security Protocols Workshop*, 2005.

[6] C. Dickens, *A Tale of Two Cities*, April 1859.

[7] R. Dingledine, N. Mathewson and P. Syverson, Tor: The second-generation onion router, *Proceedings of the Thirteenth Usenix Security Symposium*, pp. 303-320, 2004.

[8] IEEE, IEEE Standards for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, IEEE 802.3, 1985.

[9] IEEE, Information Processing Systems – Local Area Networks – Part 4: Token-Passing Bus Access Method and Physical Layer Specifications, IEEE 802.4, 1990.

[10] IEEE, IEEE Standard 802.11-1997 Information Technology – Telecommunications and Information Exchange Between Systems – Local and Metropolitan Area Networks – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11, 1997.

[11] ITU, Multifrequency Push-Button Signal Reception, Recommendation Q.24, ITU Telecommunication Standardization Sector, 1988.

[12] R. Pang and V. Paxson, A high-level programming environment for packet trace anonymization and transformation, *Proceedings of the ACM SIGCOMM Conference*, pp. 339-351, 2003.

[13] V. Paxson, Bro: A system for detecting network intruders in real time, *Computer Networks*, vol. 31(23-24), pp. 2435–2463, 1999.

[14] J. Postel (Ed.), Internet protocol, Internet Engineering Task Force RFP 791, September 1981.

[15] T. Ptacek and T. Newsham, Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection, Technical Report, Secure Networks, Inc., Calgary, Alberta, Canada, 1998.

[16] M. Reiter and A. Rubin, Crowds: Anonymity for web transactions, *ACM Transactions on Information and System Security*, vol. 1(1), pp. 66-92, 998.

[17] R. Rivest, Chaffing and winnowing: Confidentiality without encryption (theory.lcs.mit.edu/∼rivest/chaffing.txt), 1998.

[18] U. Shankar and V. Paxson, Active mapping: Resisting NIDS evasion without altering traffic, *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pp. 44-61, 2003.

[19] M. Sherr, E. Cronin, S. Clark and M. Blaze, Signaling vulnerabilities in wiretapping systems, *IEEE Security and Privacy*, pp. 24-36, November/December 2005.

[20] G. Simmons, The prisoners' problem and the subliminal channel, in *Advances in Cryptology – Proceedings of CRYPTO'83*, D. Chaum (Ed.), Plenum Press, New York, pp. 51-67, 1983.

[21] D. Song, fragroute (monkey.org/∼dugsong/fragroute), 1999.