University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS) | Department of Computer & Information Science

9-10-2010

Notes on Theoretical Limitations and Practical Vulnerabilities of Internet Surveillance Capture

Eric C. Cronin
University of Pennsylvania

Matthew A. Blaze
University of Pennsylvania, blaze@cis.upenn.edu

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Recommended Citation

Eric C. Cronin and Matthew A. Blaze, "Notes on Theoretical Limitations and Practical Vulnerabilities of Internet Surveillance Capture", . September 2010.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-10-29.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/cis_reports/939
For more information, please contact repository@pobox.upenn.edu.

Notes on Theoretical Limitations and Practical Vulnerabilities of Internet Surveillance Capture

Abstract

Surveillance of Internet communications is increasingly common. As a greater and greater percentage of communication occurs over the Internet, the desire by law enforcement, intelligence agencies, criminals, and others to access these communications grows. In recent years, motivated by updated legislation, we have seen the first large-scale systems for intercepting Internet communications deployed, and there is increasing pressure for more such systems to be developed and put to use.

Such systems raise a number of obvious questions for the security research community. Unfortunately, nearly all the systems that have been developed are closed and proprietary, and their inner workings closely guarded for commercial and "security" reasons. Very little research exists in the open academic literature exploring the technical aspects of Internet surveillance, and (to our knowledge) none which focuses on security or reliability. In this work we examine one specific problem, that of performing reliable *capture* of Internet communications.

This work has three main contributions which address some, but by no means all, of the open questions relating to reliable capture in Internet surveillance. First, we provide a survey of the current state of practice for Internet capture in the public literature. Second, we examine a number of ways in which existing capture solutions fall short of perfect capture, and the consequences, namely theoretical vulnerabilities as well as practical attacks on the accuracy and completeness of information analyzed. Finally, we construct a set of improved capture tools which provide stronger, more reliable results when used in conjunction with existing tools.

This document represents a dissertation in progress.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-10-29.

This technical report is available at ScholarlyCommons: https://repository.upenn.edu/cis_reports/939

Notes on Theoretical Limitations and Practical Vulnerabilities of Internet Surveillance Capture

Eric Cronin and Matt Blaze
{ecronin,blaze}@cis.upenn.edu
University of Pennsylvania
Department of Computer and Information Science
Tech Report MS-CIS-10-29
September 10, 2010

ABSTRACT

Surveillance of Internet communications is increasingly common. As a greater and greater percentage of communication occurs over the Internet, the desire by law enforcement, intelligence agencies, criminals, and others to access these communications grows. In recent years, motivated by updated legislation, we have seen the first large-scale systems for intercepting Internet communications deployed, and there is increasing pressure for more such systems to be developed and put to use.

Such systems raise a number of obvious questions for the security research community. Unfortunately, nearly all the systems that have been developed are closed and proprietary, and their inner workings closely guarded for commercial and "security" reasons. Very little research exists in the open academic literature exploring the technical aspects of Internet surveillance, and (to our knowledge) none which focuses on security or reliability. In this work we examine one specific problem, that of performing reliable *capture* of Internet communications.

This work has three main contributions which address some, but by no means all, of the open questions relating to reliable capture in Internet surveillance. First, we provide a survey of the current state of practice for Internet capture in the public literature. Second, we examine a number of ways in which existing capture solutions fall short of perfect capture, and the consequences, namely theoretical vulnerabilities as well as practical attacks on the accuracy and completeness of information analyzed. Finally, we construct a set of improved capture tools which provide stronger, more reliable results when used in conjunction with existing tools.

This document represents a dissertation in progress.

Chapter 1

Introduction

Surveillance of Internet communications is an increasingly common occurance. As a greater and greater percentage of communication occurs over the Internet, the desire by law enforcement, intelligence agencies, criminals, and others to access these communications grows. In recent years, motivated by updated legislation, we have seen the first large-scale systems for intercepting Internet communications deployed, and there is increasing pressure for more such systems to be developed and put to use, as well as mandating that core network infrastructure be designed to include surveillance features.

Such systems raise a number of obvious questions for the security research community. Unfortunately, nearly all the systems that have been developed are closed and proprietary, and their inner workings closely guarded for commercial and "security" reasons[1]. Very little research exists in the open academic literature exploring the technical aspects of Internet surveillance, and (to our knowledge) none exists which focuses on security properties.

With so little published foundation to build upon, it is difficult to choose where to begin. In this work we restrain ourselves in two ways: we focus on a single aspect of surveillance, *capture*, and a single aspect of capture, *reliability*, and explore the theoretical and practical aspects of Internet surveillance with respect to these. Other metrics such as quality of analysis, susceptibility to abuse, performance, and user interface are outside the scope of this work.

1.1 Electronic Surveillance

Surveillance is a broad term and encompasses any number of more specific objectives and goals when applied to the Internet: from diagnostic monitoring performed by network administrators to "wiretapping" performed by law enforcement in cooperation with service providers to spying and espionage performed completely illicitly. In this dissertation we focus on the middle case, where surveillance is performed, with aid from the network, on an uncooperative *target* (or *subject*) who may or may not be actively trying to avoid surveillance. Within the telecommunications industry and related legal community this type of surveillance is known as Lawfully Authorized Electronic Surveillance (LAES). LAES is an interesting middle ground in surveillance because, unlike diagnostic or illegal surveillance, it operates under well defined outside restraints and requirements. Whereas the specific goals and constraints of other forms of surveillance vary on a case-by-case basis, LAES is appropriate for useful general analysis.

[1] surveillance often relies, at least partially, on security by obscurity, especially for the particularly challenging task of hiding the fact that surveillance is even being conducted.

Lawful surveillance is (assumed to be) relatively well understood both from technical and legal/policy perspectives in the domain of telephone communications, where it has been performed for nearly a century. It has been commonly assumed that this knowledge and experience can be directly transfered to other forms of electronic communication such as the Internet. While this is true at the highest levels, (e.g. basic definition, abstract goals, and requirements), the technical details are quite different and not nearly straightforward. This goal of this dissertation is to explore these technical differences, but first we look at the commonalities of all electronic surveillance with the remainder of this section.

Electronic surveillance can be decomposed into four (logically) separate stages: *physical access*, *capture*, *analysis*, and *verification*. Because the surveillant is not a participant in the communications being monitored, the first step in surveillance is to gain access to the communication channel and direct a copy of all information to the next stage. The capture stage (also referred to as *collection* or *production* in some domains) is where the communications to and from the target are identified and permanently stored. Analysis takes the raw transcripts from a capture and attempts to extract from them the information exchanged between the target and their correspondant. Finally, verification takes the output of both capture and analysis stages along with information gathered through other means and reconciles them to gauge the accuracy of the surveillance. Of these four stages, physical access and capture must be performed in real-time as the observed communication occurs; analysis and verification can be performed at any point in time. A consequence of this real-time property is that any errors or imperfections in physical access or capture performed today cannot be recovered from through technological advances in the future.

Formally, for the purposes of this dissertation we define surveillance as follows: the physical access, capture, analysis, and verification of network communications between two uncooperative parties by a third party. This particular definition excludes situations such as when one of the parties cooperates in the surveillance (e.g. when a commercial service is subpoenaed) or capturing communications from any location other than the network (e.g. when keyloggers or host compromises are employed). Additionally, when we talk about *Internet* surveillance, we mean surveillance of general Internet Protocol (IP) networks. This excludes surveillance specific to particular higher-layer applications such as Voice over Internet Protocol (VoIP) or e-mail.

1.1.1 General Properties of Surveillance

Surveillance is an adversarial problem, where gauging reliability based on unlikely or uncommon byzantine failures as is done in other areas of telecommunications is insufficient. Strong guarantees of accuracy and the ability to audit and verify results are required to prevent a determined target from evading or manipulating surveillance. Additionally, because the results of surveillance are often used as the foundation for further investigation, not just as evidence in and of themselves, errors in surveillance can have cascading effects on the reliability of other evidence in the legal realm.

While the technical specifics of communications are critically important as we show in this dissertation, they are not what the surveiller is generally interested in. What the target actually sees, hears, reads, etc. is what the surveiller actually cares about, at higher “human” layers not lower network layers. For voice communication, it should be like they are listening in at the target’s ear; for computer communications, as if they are “shoulder surfing” behind the target. The point at which a communication signal goes analog as the last hop between network and the target’s sensory organs is the most accurate point to determine precisely how the target interprets communications and what information is exchanged. This is the task of the analysis stage, to transform raw electrical impulses to human understanding.

All surveillance faces the problem of *undercapture*, where communications to or from the target of surveillance are missed by surveillance. Undercapture can be a consequence of poor technical implementation,

such as failing to accurately record communications, or of incorrect application, such as performing surveillance at a position where only some of the communication is visible. When performing lawfully constrained surveillance, a second fundamental problem of *overcapture* exists. Whenever the communications channel monitored is used by parties other than the target it is the surveillant's responsibility not to observe and record these communications (intentionally or unintentionally). Again, poor technical implementation and poor deployment may both lead to overcapture, and failures in overcapture prevention can lead to undercapture of legitimate communications.

1.1.2 Requirements

Our definition of surveillance above defines the basic problem, but does not include how surveillance is evaluated or constrained. In this section, we first examine available legal and industry requirement specifications, and then define the model under which we operate in this dissertation.

Title III and FISA Requirements

In general, surveillance by any party is illegal at both the federal and local level in the United States. Two federal statutes relax this prohibition for law enforcement and intelligence agencies: Title III of the Omnibus Crime Control and Safe Streets Act of 1968 (Title III) [USC68] and the Foreign Intelligence Surveillance Act of 1978 (FISA) [USC78]. Although surveillance is increasingly conducted under FISA authorization, Title III is what is usually thought of when discussing LAES in the United States.

Title III defines two categories of wiretaps with different legal requirements for authorization: *full intercepts* (or *content* wiretaps) and *partial intercepts* (or *pen register* wiretaps). In both cases, a specific target for the wiretap order is named, and only communications in which the target participates may be monitored. Traditional wiretaps also include a specific location to be monitored, for example a phone number or ISP account; recently the courts have begun authorizing "roving" wiretaps, where surveillance is allowed anywhere the target may be communicating.

A partial wiretap contains only information *about* the target's communications, not the communications themselves. When and to whom the target communicates, whether communication was successful or not, and other "call-identifying information" are recorded for the monitored channel. A full (content) wiretap contains all this, as well as the actual contents of the communications.

CALEA Requirements

The Communications Assistance for Law Enforcement Act (CALEA) [USC94] is a U.S. law requiring the cooperation of telecommunications service providers in enabling and performing surveillance under Title III or FISA. Prior to CALEA, service providers only provided access to LEAs, while after CALEA's enactment the entire capture stage of surveillance becomes the responsibility of the service provider. The wording of CALEA is not technically specific, and while telephone-centric it applies to all forms of telecommunications, including Internet communications [FCC05, FCC06].

Sections 103 and 105 of CALEA specify several broad requirements that a service provider must meet to be considered compliant with the law [USC94, TIA02]: implementations must impose a "minimum of interference" to the subscriber's services; activation of wiretaps must require human intervention, and consequently, law enforcement should not be able to remotely instantiate wiretaps; additionally, the law enforcement agency must not have access to content or call-identifying information that is not covered under the wiretap order.

- **Intercept Categories:** There are two intercept categories of interest to [law enforcement] with respect to Broadband Intercepts: *Full Intercept* (Full Packet Data & Out-of-Band Events) and *Limited Intercept* (Packet Header Summary & Out-of-Band Events).

- **Transparency:** The Broadband Intercept must be conducted in a transparent manner, i.e., in a manner that prevents the [Subject] from detecting that an intercept is being conducted. Service parameters (e.g., bandwidth, latency, availability) must not be affected in any way by the intercept.

  The fact that an interception is being conducted must be transparent (i.e., undetectable) to all non-authorized employees of the [Internet Service Provider (ISP)], as well as to all other non-authorized persons.

  The fact that there are or may be interceptions being conducted by multiple different Law Enforcement Agencies (LEAs) on the same Subject must be transparent (i.e., undetectable) to each receiving LEA.

- **Confidentiality/Access Control:** Access to, or knowledge of, an intercept, interception capabilities, intercept-related equipment, and intercepted communications and data must be protected and limited to only authorized persons.

- **Chronology of an Intercept:** These three processes (Authentication, Validation, and Non-Repudiation) are part of the logical chronology of an intercept. Authentication is performed at the inception of an intercept to establish the connection between the communication and the [Subject]. Validation then verifies that an intercepted stream is associated with the [Subject]. Non-Repudiation confirms after the intercept is completed that the intercept was in fact associated with the [Subject].

- **Correlation:** If more than one category of intercept is active at any time for a Subject, the interception information delivered to the LEA must be accurately correlated by intercept category. [...]

  The Out-of-Band events must be accurately correlated in the intercepted information delivered to the LEA. [...]

- **Isolation:** Only communications associated with the [Subject] may be intercepted. Communications associated with the [Subject] must be isolated, and communications not associated with the [Subject] must not be captured, stored, or delivered to the LEA.

- **Proportionality:** Only the authorized communications categories may be intercepted. [...]

- **Completeness:** All communications to and from [Subject] must be intercepted for the entire period authorized by the Broadband Intercept Order.

- **Compression:** Compression must not be used in transmitting, buffering, storing, or delivering the intercepted communications to the LEA.

- **Encryption:** If the [ISP] provides encryption services to its customers or subscribers, the [ISP] must either: deliver the intercepted communications to the LEA in unencrypted form; or, provide information about the encryption algorithms used and the encryption keys to the LEA to enable the LEA to decrypt the intercepted communications.

- **Performance:** The [ISP] must be able to provision multiple simultaneous intercepts on a single Subject.

  The [ISP] must be able to provision multiple simultaneous intercepts on multiple Subjects.

  If the [ISP] requests that the LEA provide the Broadband Intercept Function, the [ISP] must provide physical facilities at the [ISP]'s premises (e.g., power, rack space) at which the LEA can co-locate the LEA-provided Broadband Intercept Function.

- **Availability and Reliability:** The [ISP] must use appropriate performance and reliability mechanisms and parameters to enable the Broadband Intercept to be performed in a manner that substantially eliminates the likelihood that the intercept will be corrupted due to dropped packets.

Figure 1.1 – CableLabs Cable Broadband Intercept Specification law enforcement requirements

In the case of content wiretaps, service providers must relay all wire and electronic communication to the LEA, either concurrently with the subject's communication or at a later date agreed to by the law enforcement agency. The service provider is not responsible for decrypting encrypted content unless the content was encrypted by the service provider. In general, the interpretation of content is the responsibility of the law enforcement agency.

Call-identifying information that is "reasonably available to the carrier" may be sent either simultaneously with content (in the case of content wiretaps) or at a later time, provided that it includes sufficient information (e.g., timestamps) to be associated with its corresponding content. When the subject of the wiretap migrates to another service provider (i.e., in the case of mobile telephony), the service provider must provide the identity of the new serving system.

CALEA specifies only what a Telecommunications Service Provider (TSP) or ISP is legally required to provide to law enforcement, not how surveillance should be implemented or how to determine if a service provider is in compliance with CALEA. To protect against this uncertainty, CALEA encourages various telecommunications standards bodies to develop specific surveillance plans for their technologies through a "safe harbor" clause: any service provider implementing one of these standards is automatically compliant.

DOCSIS CBIS Requirements

One of the few industry standards dealing with general Internet surveillance is the Cable Television Laboratories' "Cable Broadband Intercept Specification" (CBIS) [CTL09], part of the Data-Over-Cable Service Interface Specifications (DOCSIS) family of standards which all cable modem based ISPs follow. This standard acts as the CALEA safe harbor industry standard for these ISPs.

Section 5.1 of [CTL09] enumerates "law enforcement's high-level requirements" for IP surveillance; this list is reproduced in Figure 1.1. Most of these requirements clearly reflect the CALEA requirements described in Section 1.1.2, expanding on them when necessary due to the technology.

As discussed in greater detail below (**??**), the architectures of the telephone network and IP-based networks differ significantly. Although Title III and CALEA try to be technology-neutral, key concepts such as "calls" are not well defined on connectionless IP networks. This lack of direct analogues makes defining what comprises a limited intercept ambiguous. CBIS specifies Layer 4 TCP and UDP *flows* (see **??**) as the equivalent of a call, although captures still occur at Layer 2.

1.1.3 TIA JEM Report

The TIA JEM report [CAL01] examines the technical (as opposed to legal or implementation) aspects of CALEA intercepts. Although it includes discussion on generic IP, its focus is on *telecommunications* services and not *information* services as defined by CALEA. General end-to-end applications are therefore punted since they are information services. Despite this narrower focus, a number of useful observations and patterns can be extracted.

The JEM report notes that even for voice telecommunications the concept of "call-identifying information" is unclearly defined for non-circuit-based technologies. The approach taken is to identify all possible identifiers and note the complications recording or possible ambiguities in each identifier.

Carnivore [SHHPK+00] was looked at as a potential approach to collecting and filtering, but its negatives outweigh its positives (in terms of requiring less effort for TIA members) due to its generic nature. The use of first-hand information sources in private, managed packet networks (Short Message Service (SMS), CablePacket VoIP, etc) was recognized as superior to inferring from the data stream alone.

1.2 Differences Between Telephone and IP Networks

Internet is not POTS/SS7, no nice managed core. Current wiretapping techniques easily confused. Not straightforward to fix or to measure bounds of errors. Little existing research.

We assume a generic dynamically routed packet switched network, such as the Internet. Routing may be asymmetrical and change at any time. Packet delivery is a best-effort service, and loss/reordering/duplication are to be expected. (implication: what you see may not be what they get). Packets contain a source and destination field, provided by the sender of the packet. The source field is not required for correct delivery of a packet, and the sender may set it to a value other than its own identity. The destination field may also be tampered with by the sender, but the network will attempt to deliver it to the specified receiver.

Because packet metadata is untrustworthy, to assign a sender or receiver to a packet observed on a link, the best the wiretapper can do is assign the set of senders and receivers for that link/direction combination. From above, the only place both of these are not *,* is on a host link. Therefore, the interesting scenarios in this most restricted environment are when the wiretapper is on a host link. We also distinguish between an *inline* wiretapper and an *external* wiretapper based on whether or not the wiretapper is able to discern the direction of packets in the link. Because many bi-directional links in practice are constructed from a shared bus, direction is not always observable to the wiretapper.

Claim: In fact, even with additional information no other position in the network is better than being placed on the host link of both the target and correspondent together. proof sketch: being on the target's link means that outgoing evasion is impossible (baring multiple links) and the set of packets received by the target has no omissions or additions. Similarly, being on the correspondent's link means that the set of packets sent and received by the correspondent is free of omissions and additions. By combining the two sets we get a transcript containing exactly all the packets sent by the target and received by the correspondent and the reverse. This is a perfect trace at this layer, so no other position(s) can do better.

- Internet architecture
- Basic protocols, IP, TCP, UDP
- e2e routing and lack of authentication
- early and wide multiplexing
- connectionless by point of observation
- unreliable transport + implementation ambiguities

1.3 Acronyms

CALEA Communications Assistance for Law Enforcement Act

CLEC Competitive Local Exchange Carrier

COTS Commercial Off-the-Shelf

DOCSIS Data-Over-Cable Service Interface Specifications

FOIA Freedom of Information Act

ILEC	Incumbent Local Exchange Carrier
IP	Internet Protocol
ISP	Internet Service Provider
LAES	Lawfully Authorized Electronic Surveillance
LEA	Law Enforcement Agency
MAC	Media Access Control
NIC	Network Interface Card
NIDS	Network Intrusion Detection System
NIPS	Network Intrusion Prevention System
SMS	Short Message Service
TCP	Transmission Control Protocol
TSP	Telecommunications Service Provider
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol

Chapter 2

Electronic Surveillance Today

While the body of academic literature on electronic surveillance is scant, this has in no way hindered its use in practice. For both voice telephony and the Internet both official and unofficial standard practices have emerged among Law Enforcement Agencies (LEAs) and service providers. In this section we provide a survey of how Lawfully Authorized Electronic Surveillance (LAES) is performed in practice on voice telephone networks (wired and wireless) and on the Internet.

2.1 Telephone Wiretapping

Telephone wiretapping for LAES is performed in two predominant ways, depending on the modernity of the LEA and telephone network in question. The first, and oldest, is quite literally a "wire tap", a small device physically connected to the phone line of the surveillance target; the second is an automated system utilizing new surveillance capabilities built in to the equipment at the core of the phone network. These systems provide physical access and (when legally allowed) capture of audio communications, as well as recording call metadata.

The output of the capture stage, audio recordings and decoded call metadata, is already in a human-understandable form, so little to no automated analysis is needed for basic use. Once a voice recording is available, a number of analyses such as speech-to-text, speaker recognition, and voice stress analysis. These types of analysis can be performed on any voice recording, and are orthogonal to LAES.

Below, we describe the two common voice intercept architectures in more detail.

2.1.1 Loop Extender Taps

Historically, the favored approach for law enforcement wiretapping has been to tap the target's local loop. For analog wireline telephone calls, relatively little special hardware is required at the tap point; it is sufficient simply to connect a second pair of wires leading back to the LEA's facilities. To make such taps less detectable and to ensure proper isolation and level equalization of intercepted content, however, LEA use a small device called a *loop extender* at the splice point. The device copies the audio on the subject's line over to the law enforcement line, re-encodes out-of-band "loop signals", and performs level equalization of the audio.

Collection equipment at the LEA decodes the dialed digit and other call processing signals to construct a log of call events. If a full-content tap has been authorized, a recording of the audio on the loop extender line is made when the event log indicates a call is in progress. When only a limited "pen register" tap has been authorized, the event log is the only record from collection.

The first stage of analysis happens automatically as the in-band signals are decoded and logged. This sub-optimal (and vulnerable [SCCB05]) analysis *before* capture is necessitated because in-band signaling and voice cannot be reliably isolated from one another in the single analog channel; when only a limited content wiretap is authorized the original audio signaling cannot be stored. The decoded call metadata can, but rarely is, be verified after the fact by comparing it with the billing records maintained by the Telecommunications Service Provider (TSP). Any calls that fail to be recorded due to inaccurate decoding are lost forever, however.

2.1.2 CALEA Taps

The second, newer, wiretap technology was designed to comply with the U.S. 1994 Communications Assistance for Law Enforcement Act (CALEA) [USC94]. CALEA was enacted both to improve the ease and reliability of traditional loop extender wiretaps and to enable wiretaps on new technology like mobile phones where loop extenders do not work. CALEA mandates a standard interface between telephone service providers (including wireline and cellular services) and agencies that perform wiretaps. In CALEA taps, the telephone company (not the LEA) decodes the signaling information and, when a full audio intercept has been authorized, separates out the call audio to its own channel. The LEA connects to the telephone company through a standard interface, defined in J-STD-025 [ANSI03], in which the signaling information (including dialed digits, on-hook/off-hook status, and so on) and call audio are sent to the agency over separate channels. While CALEA applies only in the United States, J-STD-025-compliant switches and interception products are marketed in other countries as well.

Each LEA conducting a J-STD-025 interception leases one or more telephone lines between the agency facilities and the target's telephone switch. The first of these lines carries a *Call Data Channel (CDC)* that reports the signaling data (call times, numbers dialed, line status, etc.) associated with all lines being monitored by the agency at that switch. Additional lines to the LEA carry *Call Content Channels (CCCs)* that contain the live audio of any active monitored lines for which a full audio interception has been authorized. The CDC may carry call data for more than one active tap, and although a single CCC can carry only one call's audio at a time, a particular CCC may carry audio for different subjects at different times, with CCCs dynamically assigned as lines become active (with the assignment reported over the CDC). Although a CALEA tap may theoretically involve equipment at more than one location or even in more than one TSP (e.g. when call forwarding or other advanced features are used), the primary point of intercept, as with loop extender taps, is the target's local central office.

Once a CDC (and, when needed, one or more CCCs) has been provisioned between a switch and a LEA, installing a tap on a new line is simply a matter of configuring the CALEA delivery system at the switch to report activity on the target line. Although telephone companies are free to implement the J-STD-025 interface any way they wish, in most systems no special physical connection to the local loop of the individual target line is required. Instead, the switch itself provides physical access, connecting the target's line both to the next switching device and to law enforcement. Call metadata is obtained directly from the switching equipment in CALEA systems, and relayed separately from any call audio, allowing it to always be reliably captured with no analysis first.

2.2 Internet Wiretapping

Unlike telephone surveillance, Internet surveillance is still a new and rapidly evolving area of practice. In 2004 law enforcement agencies requested that the interpretation of CALEA be expanded to explicitly cover Internet technologies [Uni04], and in 2005 the regulating body ruled that both communication services (e.g. Voice over

Internet Protocol (VoIP)) and information services (including e-mail, the web, and instant messaging) fall under CALEA [FCC05]. At the time of this writing mandatory CALEA compliance is still a future requirement for these Internet services.

Although standardized approaches such as the CBIS standard described in Section 1.1.2 have recently been developed [CTL09, ANSI07, BFS04] for future CALEA compliance, the current state of practice is still largely ad-hoc in nature. Complicating matters further, while there are a few dozen Incumbent Local Exchange Carriers (ILECs) and Competitive Local Exchange Carriers (CLECs) providing service directly to businesses and individuals, there are thousands of independent Internet Service Providers (ISPs) operating in the United States, each with unique internal networks. This heterogeneity has led to the development of standalone surveillance systems as opposed to systems closely coupled with (and heavily dependent on) the service provider network.

As with loop extender taps for voice networks, the simplest way to provide physical access to an existing network is by placing a tap (logically) in-line on some communication link. For the optical links that are common in the Internet backbone and in the core of large local area networks, this is often a passive physical device that uses semi-reflective mirrors to split off some of the light carrying the communications [Net06]. For copper links where passive taps are impractical, powered devices emulate the behavior of an optical splitter and provide a regenerated copy of the channel to the surveillant. An alternative to physically tapping a network link is to utilize the diagnostic functionality present in most carrier-grade Layer 2 equipment. Network switches contain a feature referred to as "span ports", "roving analysis ports", or generically "port mirroring" [Cis07]. When mirroring is enabled on a given switch port a copy of every incoming or outgoing packet on that port is directed to the mirror port or a pair of mirror ports (one for each direction). It should be noted, however, that despite the similarities to in-switch access for CALEA voice taps, mirror ports are not designed for surveillance and make no guarantees of reliability [Net07, Net09, Der10].

These techniques provide an (ideally) exact physical-layer duplicate of the monitored link without altering the original traffic in any way (e.g. ordering, timing, contents, and even errors are all unchanged). This copy of is then sent either directly to a collection device, or passes through an aggregation device first and then to one or more collection devices. Aggregation devices take multiple tap inputs and feed output to multiple capture (or "sensor") devices based on simple rule sets [gig, Neta]. In addition to making point-to-point input to output connections, most aggregation devices are capable of performing simple filtering on both the input and output ports.

Capture is often performed with an off-the-shelf PC running Windows or a Unix-like operating system, with a dedicated Network Interface Card (NIC) connected to the tap output. The operating system places the capture NIC in "promiscuous" mode which makes every Layer 2 packet on the link visible to software, or uses an optimized capture-only driver for the NIC [Der08] (needed at higher bandwidths). Packets are then filtered to isolate only the authorized communications (if not already filtered by an aggregation device), and stored to disk. The unofficial standard format for permanent storage of packet captures is the `pcap` [TCP] file. A `pcap` file contains each packet (in the order of arrival) along with an arrival timestamp based on the capturing computer's clock. It is possible to capture and store less than a full packet (e.g. just the Layer 2-4 headers), but this should not be done for surveillance.

For higher capacity network links, specialized "hardware sniffer" devices [Netb] exist. These devices combine high-performance hardware-accelerated NICs with fast storage subsystems and custom user interfaces (often providing analysis features in addition to capture). Underneath the covers, though, the basic capture occurs the same, and output is almost always available in `pcap` format.

Unlike the audio recordings of telephone surveillance, `pcap` files from Internet surveillance are not very useful in their raw form to human observers. As a result, it is common for a number of different analyses to be

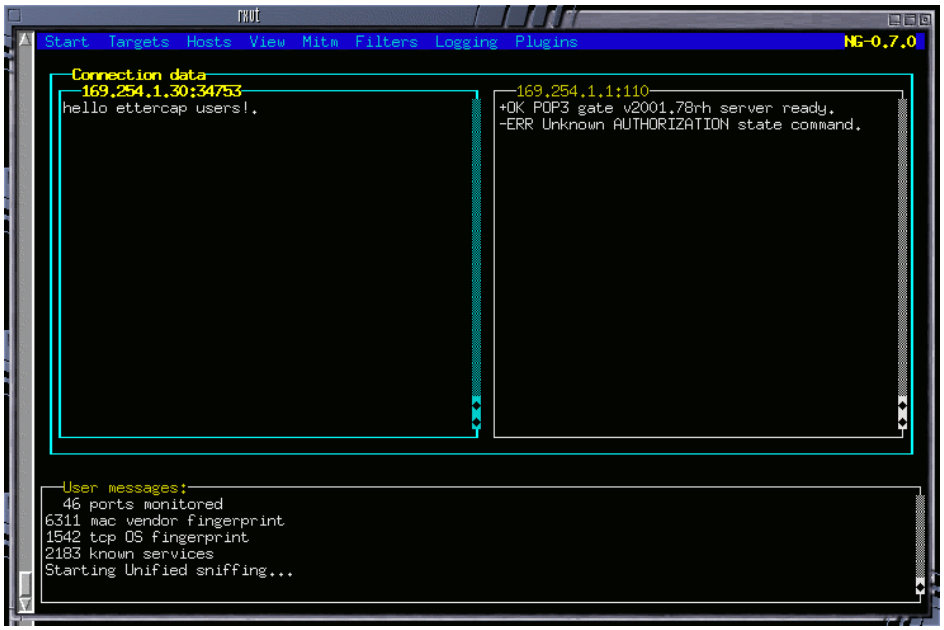

Figure 2.1 – Reconstructed TCP streams for each direction using Ettercap

performed automatically, and the results of these analyses to be treated as equal to the raw captures.

The first analysis commonly performed (and one required by the CBIS standard) is extracting flow records from the stream of packets. A flow, as introduced in **??**, is a 5-tuple of source and destination addresses, source and destination ports, and protocol (Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)). In addition to the flow signature, the timestamps for initial and final packets as well as counts of total packets and bytes in each direction are typically recorded with each flow.

Flow records provide an equivalent to call metadata for Internet communications, but the contents of the communication are still scattered across hundreds or thousands of packets. To provide a more useful transcript of communications, the second common analysis is "stream reconstruction". The abstraction provided by TCP is of two unidirectional byte-streams, one in each direction. TCP reassembly (stream reconstruction) simulates the TCP state machine in order to construct these two streams, yielding two files with each side of the communications (see Figure 2.1). More advanced stream reconstruction can maintain the timing information present in the individual Internet Protocol (IP) packets, but this is rarely done in practice. Because TCP does not synchronize the two streams and individual packets may cross in the network, reliably merging the two individual streams into a back-and-forth "conversation" is extremely difficult. Tools often provide such a view (see Figure 2.2), but it is based on the ordering of packets at the point of capture, which may not represent the order at either endpoint.

Finally, as with voice communications, more specific analyses exist, such as decoding particular application layer protocols [] or "carving" files out of reconstructed streams [Har05]. These analyses are usually performed in addition to, not in place of, the more basic types described above, and are equivalent to the more detailed analyses available to voice transcripts.

The remainder of this section describes specific software tools used for Internet surveillance.

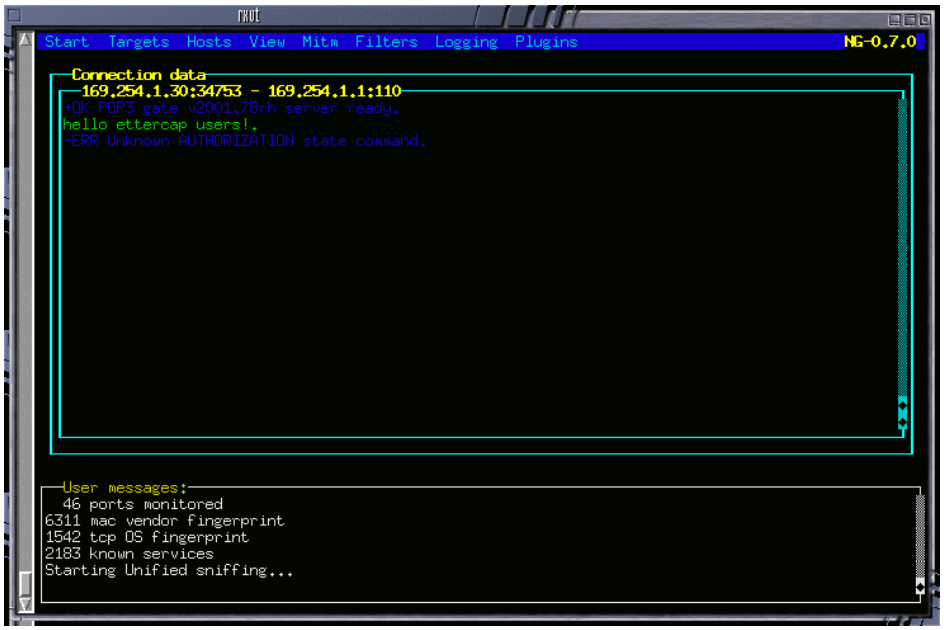

Figure 2.2 – Reconstructed TCP streams combined to show conversation using Ettercap

2.2.1 General Purpose Network Monitoring Tools

[CSVD04, QoS, Wire**?** , Ette, ngre]

The first category of software tools are general purpose "sniffers". These tools are generally interactive in nature, providing various types of analysis on-demand.

2.2.2 LAES-Specific Tools

Carnivore (DCS-3000)

In the late 1990's, the FBI developed a tool, DCS-3000 (more commonly known by the ominous code-name of one component, "Carnivore"), to perform automated surveillance [EFF06] within ISPs. This is the most thoroughly developed system whose dedicated use is performing LAES, although in the mid-2000's it was discontinued in favor of Commercial Off-the-Shelf (COTS) solutions. Information about DCS-3000 has been released to the public through evaluation reports [SHHPK+00, BBF+00] and redacted Freedom of Information Act (FOIA) releases [EPIC05, EFF06].

DCS-3000 was comprised of a capture and storage device installed at the monitored ISP, called "Carnivore", and a suite of analysis tools used inside the FBI including "Packeteer" to perform basic analysis including stream reconstruction, and "CoolMiner" to perform data mining.

OpenCALEA

For smaller ISPs, upgrading their core network to new, CALEA-enabled equipment is economically impractical. Instead, an updated version of current stand-alone surveillance systems is a much more enticing proposition. One project exists to implement such a solution using COTS hardware and open-source software: OpenCALEA [Mer09]. The OpenCALEA architecture consists of two software components which communicate over a network (either

Open Source Network Sniffers

- **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 Network Sniffers

- **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 2.3 – Network sniffer software evaluated

Commercial CALEA Solutions

[Acc, Aqs, Flu, Sol, IP , ipo, ETI, Netd, Netc, Wil, Top]

2.3 Procedural Considerations

Tools are only half the equation, how the tools are deployed and used is equally important in producing reliable results. In particular, for capture, *where* a tool is placed in the network is as important as how the tool is designed. As examined in the next chapter, the many forms of multiplexing employed in network designs significantly impact the information at a given vantage point.

Chapter 3

Modeling Communications Surveillance

A basic model for network communications surveillance is shown in Fig. 1. It consists of two communicating parties, Alice and Bob, and the surveillant Eve. The network itself is divided into three segments: The individual uplinks between Alice and Bob and the network, plus the network proper which we treat as a single cloud. The surveillant can listen in at one or more of these three locations in the network.

The purpose in dividing the network in this fashion is to take advantage of the unique properties of the two "last mile" links which are not present in the core of the network: completeness and isolation. Assuming that there is no multi-homing (multiple connections to the network core), all traffic sent and received by the end host must traverse the uplink. Within the core of the network, this may not be the case in general; for the telephone network and IP networks in particular this is almost never the case. What links within the core of the network a communication from Alice to Bob may utilize is also very dependent on the design and construction of the network, whereas the presence of all communications on the last mile is independent of network architecture. Secondly, on these last mile links there is nothing but communication to and from the end host. A surveillant constrained to only capture communications from one target does not need to perform any filtering of the communications on the uplink. Within the core of the network, however, links are almost always multiplexed in some fashion (time, space, frequency, etc.). This requires the surveillant to have knowledge of the type of multiplexing employed and careful real-time filtering to capture only the targeted communications. Even when filtering is not legally required, the surveillant must accurately demultiplex these shared links to be able to attribute communications to specific parties.

3.1 Telephone Network

Communications surveillance is generally thought of in terms of the telephone network (Plain Old Telephone Service — POTS), making it a worthwhile starting point to examine under this model. The phone network is a relatively old network at over a century in the same basic form. The service provided is fairly low-tech: a 3KHz-wide analog channel for voice communication, along with minimal signaling to establish a call and provide feedback on call progress. For the length of the call, the network provides a dedicated path between Alice and Bob (Fig. 2). Although the physical links of the network may cary many calls (thicker "trunk" lines in the figure) and use a variety of technologies (including Internet-like packetized links), at any given point in the network the unique dedicated channel between Alice and Bob can easily be separated out due to its static assignment.

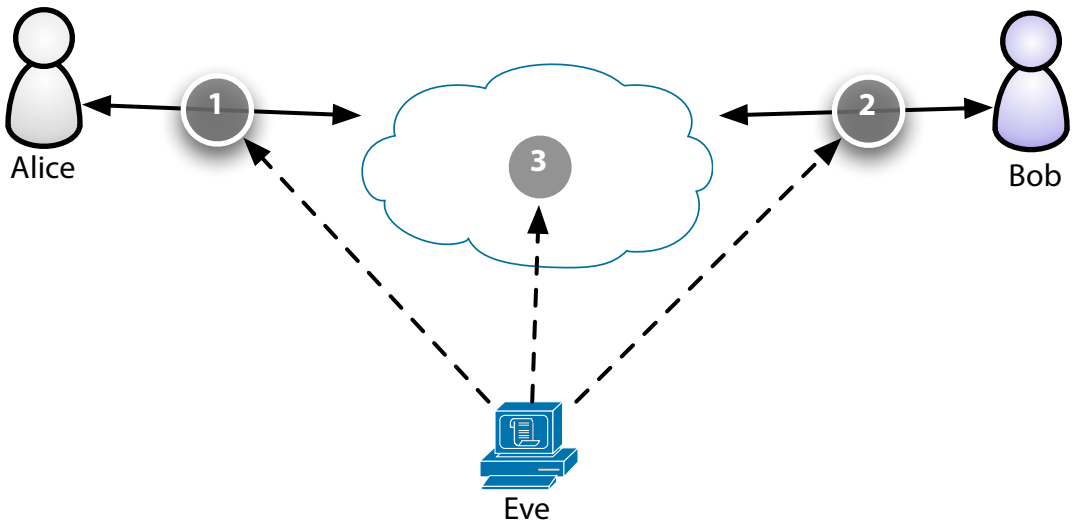

Figure 3.1 – Simplified network model for surveillance

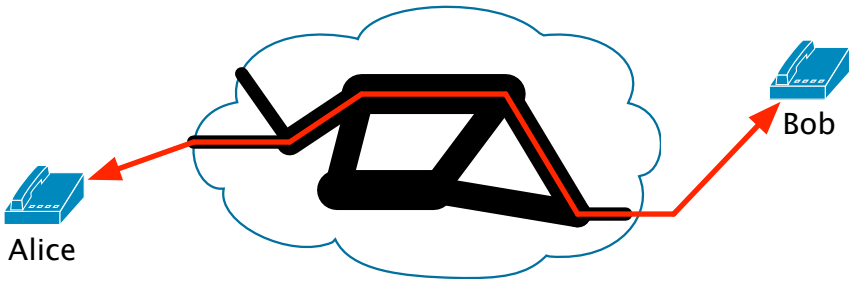

Figure 3.2 – Model of voice telephone communications network

To establish a call one endpoint communicates with the other end of its dedicated link (the "central office") and provides the address of the remote endpoint to communicate with. The network then reserves the necessary resources for the dedicated channel between the two endpoints and connects them. When either party ends the call the network then tears down the connection and reclaims the used resources. During the call itself the network performs no dynamic action.

Conclusion: position doesn't matter much in this network, and in fact core can be better than last mile (e.g. cell phone)

3.2 Internet Protocol Network

The Internet also provides a simple service, although of a very different type: best-effort connectionless delivery of datagrams.

Conclusion: position matters a lot in this network, and the core is significantly worse both for completeness AND isolation.

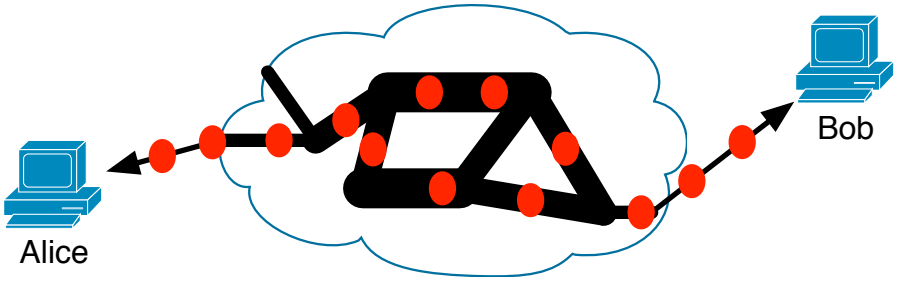

Figure 3.3 – Model of IP communications network

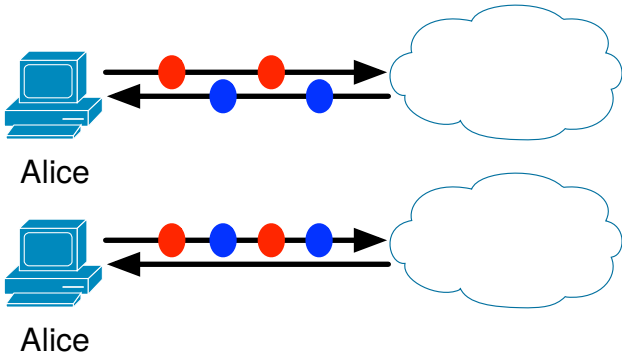

Figure 3.4 – Directionality of IP links

3.3 Last Mile Issues for Internet Surveillance

Unfortunately, modeling the dedicated last mile uplink as simply as the above model obscures several vulnerabilities as well as advantages to surveillance. Below we present three refined models of the last mile addressing individual aspects important for surveillance.

3.3.1 Directionality

Unlike an analog voice circuit where both endpoints share a single bidirectional channel, most Internet links are “full duplex”: separate unidirectional channels from A to B and B to A. This property of directionality allows otherwise identical sequences of packets to be distinguished from one another (Fig. 4). This gives an Internet surveillant an advantage lacking in the telephone network (See our work in [SCCB05] for a number of practical attacks on the telephone network based on this vulnerability).

Within the core of the network, however, it is worth nothing that the separation of traffic from A to B from traffic from B to A has a less positive influence. Due to asymmetric routing a particular physical location may only see half of the conversation. Distributed eavesdropping is likely necessary for accurate results in the core.

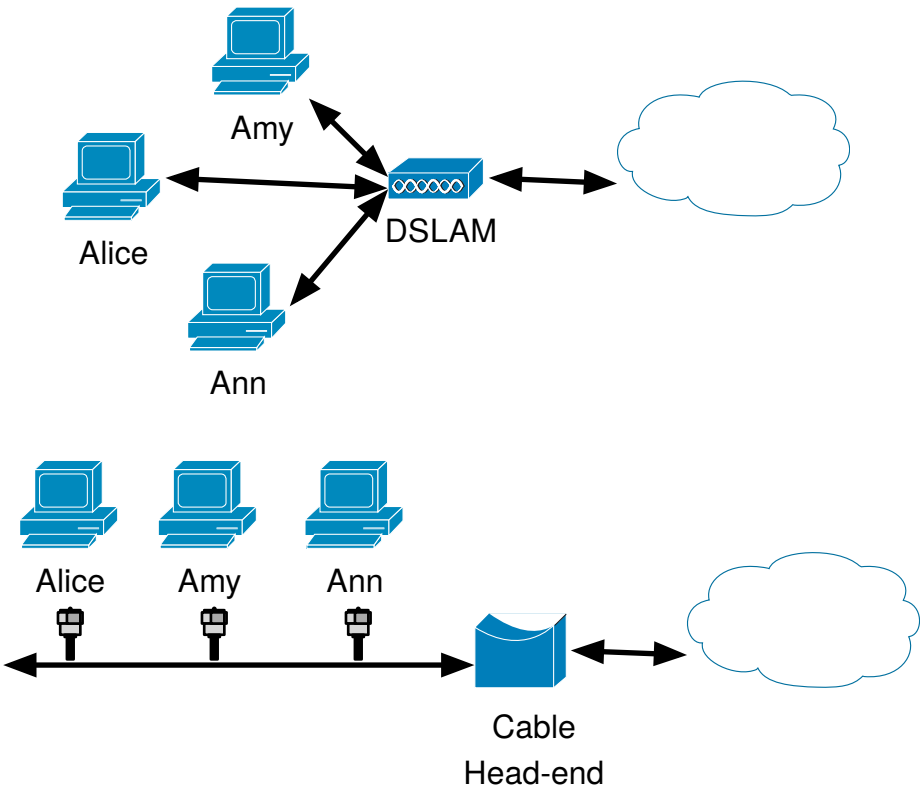

Figure 3.5 – Obscured vantage point in ISP networks

3.3.2 Vantage Point

Modern networks use a wide variety of physical- and link-layer technologies which complicate this simplistic network-layer view. Some of these technologies, such as ADSL, are very difficult for a man-in-the-middle to decode despite being point-to-point. Others, such as DOCSIS-based cable modem links, are not point-to-point at all, but instead a shared medium which has no physically isolated last-mile portion.

These properties mean that for technical or economic reasons the eavesdropper may be forced to chose a vantage point where traffic from multiple users is present.

3.3.3 Dumb Network

The previous two refinements of our basic Internet model dealt with the particular technology used in practice; the third refinement deals with a larger architectural issue with stateless, connectionless networks.

The telephone network is often referred to as being a “smart” network, while the Internet is a “dumb” network. These adjectives describe the division of work between end-host and network to provide reliable communications. In the phone network, end-to-end reliability is provided entirely by the network. The behavior of the network when operating correctly is that, with the exception of some attenuation, the signal that leaves the network is identical to the signal that enters it. The implication of this service guarantee is that position within the network plays no significant role in the fidelity of the surveillance.

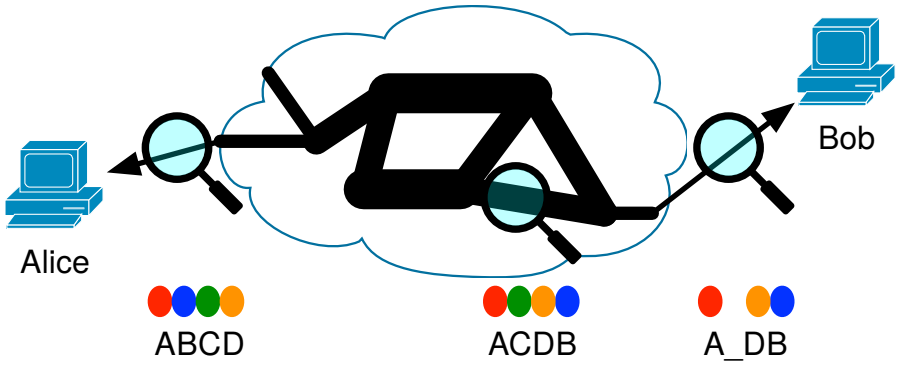

Figure 3.6 – Single vantage point inconsistencies in IP networks

The Internet, in contrast, makes almost no service guarantees and leaves all reliability to be provided by the end-host.

Chapter 4

A model of (im)perfect surveillance capture

At first glance, the capture stage of surveillance is straightforward and uninteresting, and accuracy easily measured: pick a suitable position in the network, copy the communications in a targeted channel bit-for-bit, and determine accuracy based on how successful you are at copying every bit correctly. Requirements such as minimization and engineering factors such as network multiplexing may complicate the task, but not in insurmountable ways it is generally perceived. This simplified view leads to a general model which is commonly used when discussing wiretapping, particularly for voice telecommunications networks.

Such a model, while simple and pleasant to deal with, hides a significant number of assumptions about the network being surveilled. It assumes as a starting point that distinct and distinguishable channels exist, that determining a position in the network where isolating a targeted channel from others is possible, and that the contents of the channel are interpretable with no additional information. Depending on the mode of communication, some or all of these assumptions may not hold; in the absence of these assumptions it is not obvious how the model changes: what does accuracy mean when there is no explicit channel or the meaning communication content is modulated by external forces? For the most part, this model turns out to be sufficient for dealing with traditional voice telephony, but for the Internet the communications architecture is vastly different and the basic assumptions fail to hold.

Surveillance is in essence a simulation of what the communicating hosts will actually see. The input to this simulation is the captured communication from a specific vantage point in the network. To perform this simulation accurately requires far more information than just the input however: it requires knowledge of how the middle-boxes of the network will process packets, how (unpredictable) interference from other communications will affect the targeted communications as they traverse the network, and finally how the end parties will interpret the communications. Depending on the type of network and the point of intercept, some of these variables may be easy to statically determine and therefore can be largely ignored. When these network properties are not known ahead of time and are difficult to discover dynamically, the simulation becomes inaccurate. In general, it is necessary to augment the static configuration of our network simulation with some amount of dynamic information which is not the captured communication itself, but is necessary to accurately perform simulation.

We propose a much more general model of surveillance capture called the *Context Model* based on the above observation. Precise copies of communications to and from a target remain central to this model, but we expand the capture to include *capture context*: direct or indirect metadata about the communications environment allowing the interpretation of actual communications. For voice telephony the context model and traditional model are nearly the same[1]. For the Internet, the context model allows us to reason in ways the

[1] The context model's definition of reliability predicts some of the attacks we found in [SCCB05] that the traditional model fails to

traditional model is unsuited for, and indicates how challenging a task surveillance actually is by the amount of context required.

In the remainder of this chapter we will explore the implications of this model, as well as demonstrate its usefulness when analyzing several real-world communications networks.

4.1 Accuracy in the Context Model

There is no generally agreed upon formal definition of accuracy when discussing surveillance and eavesdropping, or for capture specifically. Partially this is due to the large number of different analyses and goals a surveiller may have: even the non-formal requirements for accuracy vary widely between various scenarios.

Accuracy in the context model has at least two axes, one for the communications content itself, and one for the context. Accuracy of the content can be measured similarly to in the traditional model: how much of the target's conversation was able to be accurately duplicated and logged. The needed accuracy along this axis is largely independent of the goals of the surveillance being performed, with the exception of the case when the surveiller is not actually interested in all of the target's communications. Along the context axis, how much accuracy is required is entirely dependent on the type of analysis that will later be performed on the contents. The more precise the analysis, particularly in determining what *was* seen and not just what *may have* been seen, the more accurate the context must be.

We can define a "perfect" capture then as a capture that contains all the communications content as well as the context necessary to perform any possible analysis. For any practical communications network achieving perfection is likely impossible, but this is a useful upper limit for evaluating the complexity of surveilling a given network.

4.1.1 Roadblocks to Perfect Capture

As already noted, there are a number of roadblocks to accurate capture. This section will discuss these in greater detail. In particular, the following topics have been identified:

- For any network some context is 'entropic' — background noise from other hosts etc. — and a fundamental limitation; other context is 'structural' — topology, protocol implementation — lack of which is an implementation failure and not fundamental.
- Unreliable surveillance is a result of multiple individual inaccuracies, which combine in some manner. It is not clear how individual inaccuracies relate to one another, and wether their combination is additive, multiplicative, or some other relation.
- Certain common network architecture characteristics can have an enormous impact on reliability: layering, distributed protocols, 'smart' vs 'dumb' core, control plane architecture, etc.

4.2 The Internet and the Context Model

This section will examine the types of context necessary for interpreting Internet communication. In particular, the use of (opaque) layered communications, the 'dumb network' end-to-end model, and dynamic distributed

capture, however

routing will be examined as sources of structural context, and connectionless, packet-switched, best-effort traffic as a source of entropic context.

Chapter 5

Vulnerabilities in Internet capture

As discussed in Chapter 3, accurate surveillance depends not only on capturing the raw communications, but also on the ability to properly interpret the raw communications in the context of a large, complex, distributed environment. Chapter 3 focused on fundamental limitations, while in this chapter we examine specific practical limitations (and thus vulnerabilities) of capture in the Internet. Some of these vulnerabilities are fundamental to the Internet architecture, while others are sensitive to where and how capture is performed.

5.1 In-Band Signaling

In the Internet there is no out-of-band signaling for end to end communications. All signaling is derived from IP packets which cary both data and signaling information. Additionally, the signaling information within these packets is entirely within the control of the sending host.

5.2 Inaccurate Emulation

As mentioned in Chapter 3, surveillance interpretation essentially requires a simulation of the Internet. Even the most basic analysis involves some simulation, such as the reassembly of IP packets into TCP streams. It is very difficult to ensure that this simulation accurately emulates every aspect of the actual protocol implementations, even when no malicious activity is occurring. Ptacek and Newsham showed in [PN98] and Handley et al in [HPK01] that the TCP protocol leaves a number of edge behaviors unspecified, and that different implementations behave differently on such input.

5.3 Directionless Capture

Most capture tools and every analysis tool examined in this dissertation behave as if there is a single, directionless channel which is observed. The majority of actual physical links observed, however, are full-duplex links consisting of a pair of unidirectional channels. When separate channels are available, it is possible to determine which side of a link packets originate on without relying on untrustworthy in-band addressing. When directional information is lacking, it is possible for one side of the link to transmit bogus packets which, if legitimate would be traversing the link in the opposite direction. Although these packets can't be used to communicate with a remote host, they can be used to specifically target the surveillance, introducing irregularities or even entirely falsified communications.

5.4 Obscured Vantage Point

A single vantage point only has a limited view of the total state of the Internet. Even a perfect content capture at the surveillance vantage point is no guarantee as to the content that will actually arrive at the destination due to both structural and entropic reasons.

5.5 Inaccurate Demultiplexing

In packet-switched networks there is no way to de-multiplex traffic without inspecting the traffic itself (compare to time-division or frequency-division multiplexing). In some packet-switched networks the header used for de-multiplexing is applied by a trusted element of the network (e.g. MPLS [RVC01]), but in IP [Pos81] this information is applied by the sender. Additionally, packets in the Internet are multiplexed together implicitly by the sender, but never demultiplexed by the sender. The value in the source address field of a packet does not play a role in the delivery of the packet. Reversing this sender multiplexing is therefore prone to error and completely reliant on untrusted in-band information.

Chapter 6

Exploiting Capture Vulnerabilities In Practice

Chapter structure: describe the vulns; describe the experimental setup (network, capture, tools, eval); experiment results

6.0.1 Experimental Network

Virtual network constructed using VirtualBox [**?**] and pfSense [**?**] for routers.

6.0.2 Tools Evaluated

We chose a representative subset of the available open source and commercial demonstration version of currently available tools. These included both general-purpose "sniffers" (CommView, Ethereal, Ettercap, PIAFCTM, and Sniff-em) and Law Enforcement Agency (LEA)-oriented tools (OpenCALEA) as well as tools which can be repurposed to provide some surveillance functionality (Bro and Snort). The capabilities of each of these tools are listed in Table 6.1.

6.1 Confusion and Evasion

In [CSB06] and [CSB08] we presented several attacks against common general purpose network sniffers. These attacks used lack of information on network topology to confuse the automated reconstruction of TCP streams. Below we summarize the attacks and results from this earlier work, and then present the planned extensions for this thesis.

6.1.1 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.

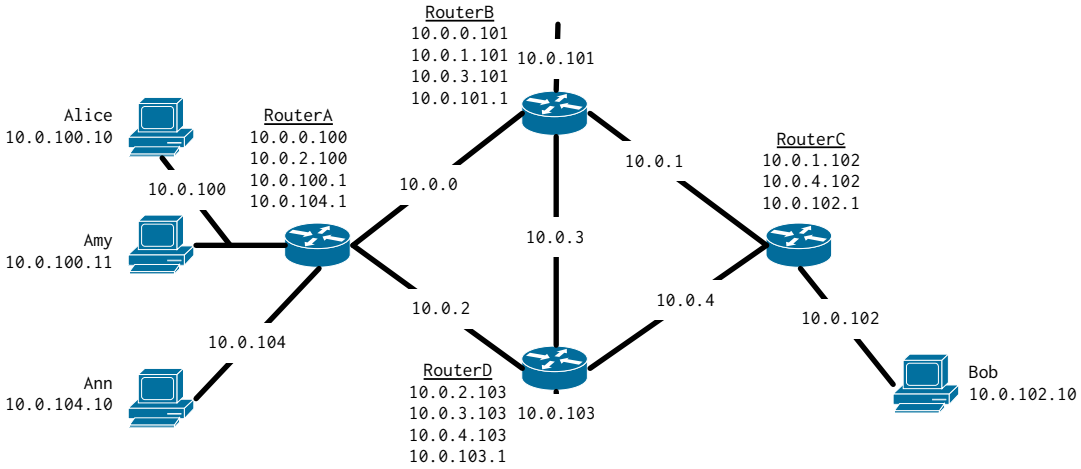

Figure 6.1 – Topology of experimental network.

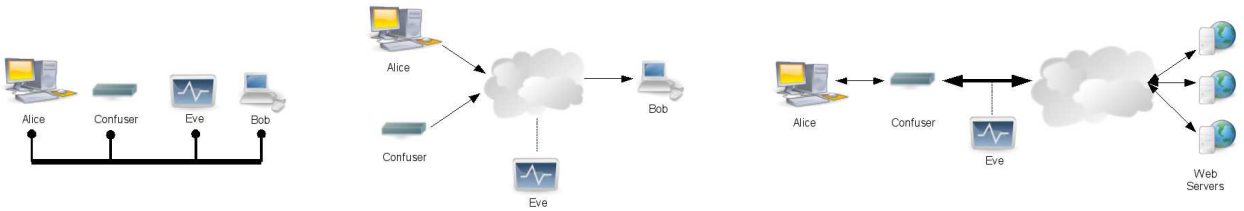

Figure 6.2 – Left: An eavesdropping topology in which all parties communicate via the same shared bus. Right: A configuration in which Eve is located on the network between Alice and Bob.

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.

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 [IEEE05, IEEE90, IEEE03]. 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 6.2 (*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.

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.

As we show empirically in Section 6.2, 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 be reliable unless it both intercepts and processes link layer headers.

Network Layer Confusion

If Eve intercepts a packet on the path from Alice and Bob (see Figure 6.2, *right*), 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 6.2, 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.

6.2 Failure of Current Eavesdropping Systems

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

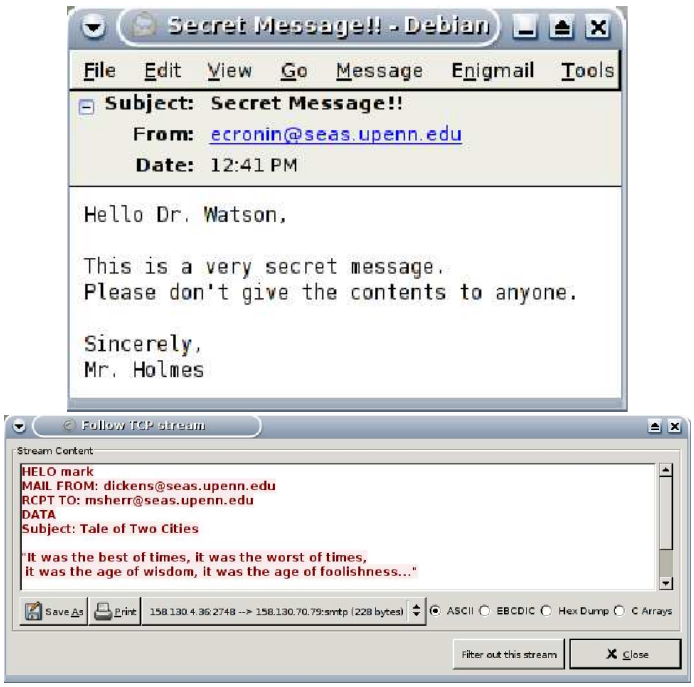

Figure 6.3 – Top: Legitimate message received by the SMTP server (Bob) and the intended email recipient. Bottom: An eavesdropping system's (Ethereal) reconstruction in which Eve not only fails to capture Alice's message, she also perceives the covertext as the legitimate message.

Digital eavesdropping evasion

To demonstrate the susceptibility of current eavesdropping tools to confusion, we implemented the MAC and TTL confusion techniques described in Section 6.1.1 and originally introduced as NIDS attacks in [PN98]. (Fragroute [Son] 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 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" [**?**]). 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 **??**). 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. Table 6.2 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 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 6.3). In all cases, the email server (Bob) correctly received Alice's communication and delivered the email to its intended recipient.

6.3 Session Replay

As discussed in Chapter 3, a lack of directional information for a channel being captured places the surveillant in a vulnerable position. In [SCCB05] we showed how a surveillance target could easily forge both the content and call-identifying information for a voice phone call unilaterally by exploiting the directionless nature of voice telephony. An analogous attack against Internet surveillance is possible when the collection point is on the same (logical) Layer 2 segment as the target. Because many consumer Internet connections employ some form of bridging or lower-layer tunneling (e.g. DSL bridging, PPPOE), the logical segment may extend into the service providor's facilities where collection often occurs.

When this precondition is met, the target can pretend to be both sides of a fake communication which will be captured by the surveillant. Sending packets to the fake remote endpoint are easy since the network does that normally. Packets from the fake endpoint to the target are more challenging, as the IP routing will expect them to be travelling in the opposite direction, and they will not propagate far. However, because the target and surveillant share the same segment at the link layer, it is possible to direct these forged packets to the IP next hop using link layer addressing and as a consequence have them be captured.

6.4 Neighbor Impersonation

Because accurately demultiplexing IP traffic by sender is, except in very specific circumstances, impossible once multiple links have combined, an obvious way to improve accuracy is to position capture prior to any multiplexing. Unfortunately, for many network technologies this is technically impossible. We look at the two predomanant consumer Internet last-mile technologies, DSL and DOCSIS cable, and show that for both it is technically and/or economically impractical to position a capture point in a position before some multiplexing has occured. In DOCSIS, the vulnerability is a result of the technology: every customer sharing a head-end is physically connected to the same cable segment. Multiplexing occurs within the cable modem itself, and forging source MAC addresses for a cable modem is harder than forging source IP addresses, the tools and knowledge to do so are widely available [Der06]. In DSL, although a point-to-point un-shared channel exists between the target and local telephone office, it is terminated into a DSLAM shared with a number of other customers. DSLAM equipment is designed to terminate and bridge traffic efficiently, not to perform

surveillance. Because there is no operational need to segregate traffic from each incoming line, the DSLAM equipment we have studied does not do so. The earliest intercept point is on the uplink leaving the DSLAM, which contains traffic from every line terminating there. Placing an capture device on the DSL line between target and DSLAM is a possibility, but because of the complex analog signalling used by DSL such a capture is technically challenging and economically unatractive.

Table 6.1 – Surveillance tools evaluated and their capabilities

Software	No Confusion (1B pkts)		MAC Confusion		TTL Confusion	
	Inter-pretation	**Detected Anomalies**	**Inter-pretation**	**Detected Anomalies**	**Inter-pretation**	**Detected Anomalies**
bro	Success	None reported	Failure (Covertext)	Retrans. in-consistency	Failure (Covertext)	Retrans. in-consistency
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 application incorrectly interpreted the covertext as the legitimate messagetext. See Figure 6.3.
Failure (Random noise) - No discernible English text could be obtained from the eavesdropper's interpretation.

Table 6.2 – Ineffectiveness of various eavesdropping software against confusion techniques.

Chapter 7

Augmented Capture and Capture-aware Analysis

The previous chapter demonstrated several exploitable vulnerabilities in Internet surveillance due to reliance on untrustworthy or unknown capture metadata: one based on the claimed communicating identities, one based on the network topology, and one based on a combination of both.

The attacks in Chapter 6 come about from two main causes: fundamentall vulnerabilities in single-vantage-point capture, and implementation vulnerabilities in packet-only capture. We propose to address these vulnerabilites in two ways, first by recording more than just the raw packet stream at the point of capture (and being careful to not discard any available metadata such as lower layers or directionality), and second by using metadata from other vantage points to guide the analysis of captures.

7.1 Discarded/Missing Metadata

7.2 Tool-Agnostic Solution

The most straightforward approach to fixing these vulnerabilities in a given tool would be to modify the tool itself. Unfortunately, as we showed in Chapter 2, there are a large number of tools in use, and many of these tools are not open-source or even available to non-law enforcement users. Modifying each (available) tool individually is therefore an unpractical as well as incomplete solution to the problem. Instead, we propose a system which sits in front of existing, unmodified analysis tools, and (non-destructively) manipulates the captured packet stream input to these tools in order to prevent attacks.

As discussed in Chapter 2, the common format for full Internet captures is `pcap`. Most specialized capture tools have an option to output as `pcap` traces, and most analysis tools, both open source and commercial, accept `pcap` traces as input. Few analysis tools, however, support additional inputs containing context for analysis or additional traces from different vantage points. For tools that do accept some context, there is no standardized way of providing it. Because of this, the approach which leads to the greatest flexibility is to provide a set of pre-analysis tools which take pcap traces and context as input, and generate a new pcap trace as output. Some of these tools may act as filters (for example using topological information to remove packets with insufficient TTLs), while others may combine multiple traces in a safe way.

The NetDude [Kre] project has resulted in several libraries useful for manipulating `pcap` traces: `libnetdude` and `libpcapnav`. Using these libraries it is possible to construct filters of the sort we desire. While

the input and output of `libnetdude` is still a standard `pcap` file, internally packets can have arbitrary additional state attached to them, and `libpcapnav` makes processing traces easy and efficient.

7.2.1 Filter 1: Hop-count

The first filter allows the surveillant to compensate for the difference between their capture vantage point and the position in the network of the actual receiver. The input trace is filtered to remove any packets whose Time-To-Live value prevents them from having reached the receiver a certain distance away.

7.2.2 Filter 2: MAC::IP Binding

When a capture is made on a network link carrying traffic from multiple sources the surveillant must demultiplex the traffic in order to perform accurate analysis. For Internet captures this means separating traffic based on the source Internet Protocol (IP) address, which may or may not reflect the actual sender of the traffic. In the last-mile of the network, more reliable link-layer addressing is often visible from the cable or DSL modem used. This filter processes the input trace and splits the traffic based on a MAC address and not the IP address.

7.2.3 Filter 3: Directional Merge

When a bi-directional channel is properly captured into two unidirectional streams it is necessary to merge these into a single trace for legacy tools to be able to see both sides of the conversation. Instead of blindly merging, the directional merge filter tracks the IP sources and destinations it has seen in each direction, and only merges automatically if there are none which appear in both directions. When ambiguities are detected, the filter can be provided with the correct direction in order to discard bogus packets.

7.3 Evaluation of Filtered Captures

The experiments from Chapter 6 were performed a second time after first processing the input with the appropriate pcapfilter tool.

Chapter 8

Related Work

In addition to the surveillance techniques looked at in Chapter 2, there are a number of research areas which overlap with the problems (and solutions) studied in this dissertation. Network Intrusion Detection Systems (NIDSs) and other forms of deep packet inspection face many of the challenges with reliability that surveillance does. The challenge of identifying the true source of Internet communications has been widely acknowledged, and a body of literature exists in this area. Surveillance is just one investigative tool used on computers and computer networks; forensic approaches are also employed both preventatively and retroactively. Below we examine these related fields of research.

8.1 Deep Packet Inspection

Surveillance is one of a number of tasks that can be classified under the broad umbrella of "Deep Packet Inspection" (DPI), where packet headers and payloads are captured at some point in the network and then analyzed and acted upon. Related tasks such as network intrusion detection, protocol analysis, regulatory/policy compliance, and performance analysis overlap in varying degrees with surveillance, differing mostly in the types of analysis performed. Of these related tasks, only intrusion detection has seen significant study in the academic literature; the remainder are represented by a variety of closed, commercial solutions with little documentation of their inner workings.

8.1.1 Network Intrusion Detection

The requirements for capture overlap a great deal between surveillance and network intrusion detection. The main differences are that the type of analysis performed by NIDSs is constrained compared to that of surveillance, while the visibility, passive/active nature, and position of a surveillance system are constrained compared to those of a NIDS. Additionally, the economics of both differ greatly, whereas the cost of a NIDS is amortized over and entire network of hosts, a surveillance system is generally dedicated to a single target.

Intrusion detection systems (both host-based and network-based), as well as the recent active variants marketed as Network Intrusion Prevention Systems (NIPSs), focus on identification (and blocking) of malicious traffic directed towards a protected network. Like surveillance, network intrusion detection/protection requires careful analysis of network traffic. Two separate aspects of NIDS/NIPS research are directly related to the problems explored in this dissertation: algorithms and techniques for accurately capturing and processing network traffic, and metrics for evaluating and comparing the accuracy of systems.

The primary goal of a NIDS is to identify malicious traffic, which can have false negatives (undetected

malicious traffic) as well as false positives (benign traffic detected as malicious). In modern NIDS what makes traffic malicious or not is a complicated and evolving question, which depends on both the contents of the data and statistical properties of the data and metadata such as the rate of packets or the correlation of port numbers in Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) packets. Packets are infrequently saved permanently, instead "events" summarizing one or more malicious packets are the main output of a NIDS. This allows optimizations to the capture engine and encourages closely coupling capture with analysis.

Because a NIDS is operated by the owner of a network, it is not constrained in many the ways a surveillance system is. Minimization is not a requirement, and the network is relatively easily partitioned into "inside" and "outside". Complete transparency is also not a requirement [SP03], and, in the case of intrusion prevention, active modification or dropping of traffic is the expected behavior. Likewise, it is acceptable for a NIDS to perform "normalization", altering forwarded traffic to remove ambiguities from analysis [HPK01]. Because NIDS and NIPS are defensive technologies where fail-safe (e.g. blocking all traffic) is preferable, they are often deployed in-line in critical network choke points. Real-time performance, on the other hand, is critically important to intrusion detection, whereas there is some leway for offline analysis of evidence gathered by surveillance. This drives much research in the area to investigate high-performance detection where some decrease in accuracy or flexibility is undesirable but preferred to buckling under load.

NIDSs/NIPSs are highly constrained in where they are located, as they operate within a single administrative domain. Necessarily, they tend to be positioned closer to the "trusted" hosts within the administrative domain than to any malicious hosts on the Internet. As shown in this dissertation, due to multiplexing surveillance is most accurate when positioned as close as possible to the target.

Intrusion detection systems are evaluated by comparing their behavior to that of an "ideal" intrusion detector on a particular input [Den87, DCW+99, LFG+00, LX01, GU01, GFD+06, GFD+05]. In order to determine the ideal behavior, this input is typically artificially generated by inserting malicious traffic into a benign trace. Different metrics have been proposed to quantify the difference in behavior between particular IDS systems and the ideal case, primarily examining the false positives (intrusions in the experimental results but not the ideal results) and false negatives (intrusions in the ideal results but not the experimental results) [Axe00].

Such metrics may be adaptable to surveillance, with omissions and insertions playing the role of false negatives and positives. Unlike intrusion detection, however, the costs, impact, and meaning of omissions and insertions are less clear. Additionally, the reliance on the quality of the evaluation input is a significant limitation to these evaluation techniques as opposed to more formal claims of accuracy.

From the offensive side, there is a body of work on techniques for bypassing NIDSs, usually by defeating the analysis and detection stages, not the capture stages. The canonical references for NIDS vulnerabilities are[Pax99, PN98], which exploit ambiguities in the Internet Protocol (IP) and TCP protocols to disrupt analysis. Vulnerabilities exploited by NIDS attacks are often also present in surveillance systems, and in fact some of the attacks in Chapter 6 are based on observations from [Pax99] and [PN98].

8.2 Computer Forensics

Surveillance is only of use *prospectively*, that is, when implemented before the desired communications occur. When investigating events in the past, *retrospective* computer forensic approaches must be relied upon. Research in this area looks both at the types of forensic analyses available and at the ways in which networks can make forensic analysis easier.

8.2.1 Forensic Analysis

After a computer attack has been detected, a forensic analysis is often performed to learn how the attack occurred and what the attacker did with the compromised resources. Analysis of available network traces is often a part of this analysis, but because analysis begins after the attack has actually occurred, capture must have been carried out ahead of time. This means that forensic analysis is often dealing with incomplete and sub-optimal network captures.

A number of guides have been developed for "first-responders" to computer attacks [Tec01, Com09], detailing how to preserve any evidence that might remain, including network logs. Some of these guides also discuss how to perform rudimentary manual analysis of network events [Tec04, Cas04, Joh06, KKKM06].

As opposed to the automated nature of surveillance, the forensic process is tailored much more to the evidence available on a case-by-case basis. Several well-known and interesting attacks and their analysis have been documented by those involved [Spa89, SM96, Sto89].

8.2.2 Forensic Logging

Like intrusion detection, another defensive strategy some network administrators utilize is the forensic logging of traffic on their network to provide better data to possible future investigations [SMSB03]. These logs can then be examined once a host is found to be compromised, an accusation of launching an attack is made, or even to troubleshoot performance issues on the network.

Several projects have been proposed to archive all traffic within a single organization (or all traffic entering and leaving the organization) in a "Packet Vault" [AUH99, ACF01, IS03, KPD+05]. The emphasis of this research is on the practical, secure and economical storage of traffic, not on the capture itself. The vulnerabilities discussed in this dissertation apply to these types of systems as well.

Because of the massive storage requirements of indefinitely archiving every packet traversing a network, in practice either a subset of raw packets are kept, or records summarizing many packets are recorded instead (or both). Because Internet traffic is dominated by TCP and UDP traffic, it is common to record "flows" instead of individual packets [CSVD04, QoS, HBP06]. On busy networks, even logging every flow is impractical, and further aggregation is performed [Cact, NTOP].

Because this logging is performed by the same organization being monitored, the minimization oversight is usually fairly limited. Accuracy to the degree of a NIDS or wiretap system is also not a key design metric, particularly the unexpected behavior induced when deliberately evading or inserting or using non-standard protocols.

8.3 Participant Identification

A key requirement for surveillance is the ability to reliably identify the participants of a communication. Both defining and determining "identity" with respect to Internet communications is a challenging problem as already seen. Attributing a given packet to a particular network, host, application or person in the general case is an open problem. Research has been conducted in both areas.

8.3.1 "Identity" on the Internet

Ideally, when performing communications surveillance the actual human beings at each end of the channel are the identities of interest in identifying. However, these humans sit several layers abstracted from what is actually observed in the network. Instead, a more network-centric definition is usually used; in this dissertation

we have informally assumed identity to be the computer(s) associated with the customer end of a particular Layer 2 link. Deeper in the network, the only viable identifier is the Layer 3 IP address, which as we have already seen is unreliable and easily forged.

For mobile devices on unauthenticated networks (e.g. campus wired/wireless networks), the Layer 2 Media Access Control (MAC) address, visible to the switches and wireless basestations, has been proposed as a static identifier which (theoretically) uniquely identifies a particular hardware device [ETSI07]. Unfortunately, because MAC addresses are easily changed and only globally visible, this type of identification is easily defeated [BPA04]

The most thorough examination of this area is Richard Clayton's dissertation [Cla05], which looks at the problem of translating an IP address into a person. The (negative) results of this work in the general case motivate both our definition of identity (which lies partway between an IP address and a person) and our model differentiating the "last mile" from the network core (Chapter 3).

8.3.2 Packet Source Identification

As we argue in Chapter 3, attributing an intercepted packet to a sending (or receiving) host reliably using the contents of the packet alone is impossible on the Internet. In this dissertation we solve this by restricting the position of the surveillant, gaining additional information from the network topology. Other solutions have been proposed in the literature for attacking this challenge which we summarize below.

Router-aided Sender Identification

Another obvious approach with significant technical hurdles to practical implementation is to log each packet that passes through a router [Sag98]. This avoids problems caused by modifying the packet itself, and allows tracing at the granularity of a single packet, but requires unscalable storage resources at every router as well as knowledge of which routers to query when performing traceback. The performance limitations are the most critical, despite the fact that logging can occur in parallel with routing e.g. using a tap: with multiple ports running at 10 Gbps, routers would require several terabytes of memory (with 10 Gbps of memory bandwidth) to store a one minute rolling log of packets. Because network speeds have historically grown faster than high-speed memory sizes and bandwidths, this limitation is likely not to improve with time. Such raw packet logs are also a tempting target for attackers, since malicious and non-malicious packets are archived alike.

To reduce the memory requirements, one could log packets probabilistically instead. The sampling rate can by tweaked to fit the trends between interface speeds and storage speeds at a particular time. This approach is the dual of the simple single-address probabilistic packet marking scheme in [SWKA01], and causes similar complications. Tracing a single packet's precise path is impossible since it is unlikely it will have been logged at every hop in its actual path. With a set of packets to trace assembling a list of routers is possible, but would require probing each router for each packet, and would require post-processing to construct paths from individual routers. Unlike any of the previous schemes, however, there are no false positives with this approach. If a router has a particular packet in its log, then there is no possibility that it was not present at that point in the Internet.

A more scalable solution proposed in [SPS+02] stores only a hash of each packet, and uses Bloom filters [Blo70] to further compress the logs. This approach introduces a risk of false positives, but by storing a record of every packet forwarded eliminates false negatives. By carefully tuning the Bloom filter parameters the false positive rate can be kept very low.

Building on this efficient packet logging technique at individual routers, Source Path Isolation Engine (SPIE) is an architecture for performing automated traceback across multiple networks. The component of each

router which digests, hashes, and stores packet records is called the Data Generation Agent (DGA). Each stores the current Bloom filter as well as some number of recent filters (along with metadata such as the hash functions used and the timespan the filter covers). One or more more powerful hosts with long-term storage known as SPIE Collection and Reduction Agents (SCARs) manage several DGAs for a region of an ISP's network. The SCAR is responsible for archiving the logs from its DGAs along with information about the local topology and routing state. The final component of the SPIE architecture is the SPIE Traceback Manager (STM), which is the externally visible interface for each ISP. When the STM receives a request to trace a packet, it distributes the digest and rough time windows to the SCARs, which then combine their archived topology with Bloom filter query results to produce a subgraph for each SCAR. Finally, the STM merges the subgraphs and returns the final results to the requester.

Unlike packet marking, SPIE relies on an online service which has the potential for abuse and denial-of-service if publicly available. Because a query can only be performed on an entire packet digest (not just "all packets from host A"), the architecture inherently limits successful queries to those who possess the original packet or its digest. This includes both the victim, as well as any intermediate router or third-party observer (e.g. law enforcement). Although the design of the DGA is such that performance is not impacted by queries to SPIE, the SCARs and STM can be overloaded if asked to perform too many traces in parallel. To protect against this, the authors propose strong authentication and only allowing trusted peers to initiate SPIE queries.

Of the three traceback approaches, SPIE has the potential to provide the most accurate results at the finest granularity. An additional feature discussed in [SPS+02] is using the SCARs to also manage Transform Lookup Tables (TLTs), mapping one digest to a different digest when a transformation like IP fragmentation, NAT translation, ICMP reply, or tunneling is performed within the ISPs network. Because the traceback data is both generated and stored within the ISPs core and can be examined retrospectively the forensic strength is stronger than for packet marking's implicit authentication. Requirements for incremental deployment of DGAs are similar as for packet marking engines, although the memory speed and size requirements for SPIE are much closer to the current technological limits than the processor requirements for packet marking are.

The major limitation to SPIE is that it does not solve the political and administrative issues present in coordinating traces through multiple provider's networks. Cooperation is needed both in widely deploying DGAs in most ISPs, but in allowing automated queries of each-other's STMs for every trace performed. SPIE also does not handle the situation when a network in the middle of the path does not participate well. To proceed beyond such a point it is necessary to determine what networks can route traffic through the missing network (keeping in mind that such peerings may be unadvertised and asymmetric), and then restart the trace at the STM of each of those networks.

Packet Marking

The idea of each packet possessing a record of its journey was present in the development of IP, evidenced by the IP record route extension [Pos81]. This extension is intended to be enabled by the sender, however, which an attacker seeking anonymity is unlikely to do. Additionally, it and other simple route recording proposals have the negative effect of causing the packet to grow in size with each hop. Every Layer 2 data link protocol has a maximum segment size, and when IP packets exceed this they must be split into multiple fragments. Fragmentation is damaging to both performance and reliability, and causing it to occur on a regular basis would be unacceptable.

The first paper to explore practical techniques for packet marking was by Savage et al in 2000 [SWKA01]. Unlike [BC00], the authors of [SWKA01] assume that while real-time cooperation among administrators is still impractical, pre-deployment of an automated marking solution at most routers is feasible. Each router

maintains its autonomy, and there is no traceback-related communication between the victim and routers.

As noted, simply appending each router's address to a list in each packet does not work. If the list size is dynamic, then packets can grow arbitrarily large causing fragmentation. If the list size is fixed, the attacker can inject packets with the list already filled causing legitimate routers not to append their addresses. If the fixed-size list is FIFO, then an attacker further away than the list size will be untraceable.

Instead of deterministically marking every packet, routers can instead mark a fraction of the packets it forwards. It turns out that when probabilistic marking is used, a single address field in each packet is sufficient to reassemble a single attack path if all routers use the same probability of marking p. The probability that a router d hops from the victim will mark a given packet is p, while the probability that each of the $d-1$ routers between it and the victim will *not* mark it is $(1-p)^{d-1}$. The probability that the victim receives a packet marked by the router d hops away is $p(1-p)^{d-1}$, which monotonically decreases as d increases as long as $p > 0.5$[1]. The reconstructed path is obtained by ranking the frequency of each router in the captured traffic.

Although simple, efficient (assuming packets can be marked without causing fragmentation), and robust, this scheme requires massive amounts of traffic in order to produce accurate results. With $p = 0.51$, the expected number of packets received before the first containing the address of the router 15 hops away is $42,000$. To rank the routers statistically significantly requires an order of magnitude more traffic be observed.

Instead of implicitly determining the depth of a given router by its frequency, the final marking solution arrived at in [SWKA01] does so explicitly by adding a distance counter that routers increment before forwarding. Additionally, instead of a single address, each packet carries an edge, consisting of two (virtually) adjacent routers: *start* and *end*.

As before, each router with probability p decides to mark or not mark a packet before forwarding. If it decides to mark the packet, it places its address in start, clears the address in end, and sets distance to 0. Otherwise, it increments distance, and if the old value was 0 places its address in end. As long as every router increments distance (and saturating addition is used), it is impossible for the attacker to provide a (start,end) pair which is closer to the victim than all legitimate edges. With this scheme, p can be much smaller, and the attack path converges much more rapidly. Additionally, only a single packet containing each edge is required, not a statistically significant number as before. The expected number of packets received before the first with an edge of depth d is less than $\frac{\ln(d)}{p(1-p)^{d-1}}$. Setting $p = 1/d_{max}$ gives near optimal results for paths up to d_{max} hops. In the example above, when $d = 15$ and $p = 1/15$, fewer than 110 packets are required on average to discover all 15 links. Because the start and end of two edges can be correlated to construct the attack tree, edge-based schemes are able to trace attacks from multiple sources without requiring pre-computed network maps or probing to discover edges from individual nodes.

Although the space requirements are less than for full routes, edge-marking still requires storing 72 bits of data (two 32-bit IP addresses and one 8-bit distance counter) in each packet. Because evolving the standard fields of IP to make such fields mandatory is infeasible, and resizing existing packets to add them impractical, the authors look for redundant information in both the 72-bit edge mark and the IP header to allow compression and multiplexing. The solution used in the paper is a combination of compressing the start and end fields into a single 32-bit *edge-id* using XOR, and splitting each edge-id into k smaller fragments and adding an offset field. The fragment, offset, and distance value are stored in the 16 bit identification field of the IP header. This encoding increases the number of packets required to reconstruct the path log-linearly in k. For the exact encoding used in [SWKA01], which splits each edge-id into 8 fragments, reconstructing a 15 hop path with 95% accuracy would require fewer than $2,000$ packets. When multiple attack sources exist, reassembly must try all possible combinations of fragments at each depth. This causes the cost of computing paths grow

[1] If $p < 0.5$ for non-malicious routers the attacker can insert inaccurate nodes in the traceback by setting the initial value of the mark field

exponentially.

Because it alters IP headers in unexpected ways, implementing packet marking without a dedicated header field conflicts with IPSec [KS05] as well as any other protocol which either depends on certain IP headers not being modified, or modifies them itself in conflicting ways. Only a single version of this scheme could be deployed on the Internet using such "hacks". Additionally, the identification field has a legitimate intended use, namely allowing IP fragmentation. A fallback technique allowing traceback to still be performed when fragmentation is present greatly complicates both marking and reassembly, and increases the performance cost. A second limitation of any packet marking scheme is that just like the source address the router or edge field contained in a packet is unauthenticated and the attacker choses its initial value. The closer the attacker is to the victim and the lower p is, the more likely that packets will arrive without an intermediate router marking the packet. The distance counter prevents creating false edges closer than the attacker, but an entire fake network beyond the final traceback-enabled router can be constructed. Determining the valid suffix in these situations requires either active verification (e.g. traceroute) or an Internet map as in [BC00]. In [SP01] an alternative, authenticated marking strategy is proposed, allowing post-hoc verification that the edge-id contained in a packet was placed there by the router at start and not by the attacker. When using unauthenticated edge-ids for forensic purposes, it is very difficult to be certain of the valid suffix even though the traceback computation can be performed offline and long after an attack concludes. A second forensic restriction is that any individual packet can only be traced (probabilistically if edge-ids are fragmented) to a particular link. If multiple attackers are present, attributing a packet to a single attacker with certainty is not possible.

An interesting extension of packet marking is proposed as future work at the end of [SWKA01]. In many DoS attacks, the forged packets are not sent directly to the victim. Instead, a packet is forged with the victim as the source address towards a service with a large amplification factor (e.g. the same traffic generators used to carry out active probing above). These services are not themselves malicious, but are unwittingly abused; until notified (manually) by the victim they have no reason to suspect requests are forged. When services like this cannot simply be disabled (e.g. DNS servers), a possible defense would be to copy the edge-id from each request, and return it as part of the reply. A victim receiving unexpected packets from these services would then be able to perform traceback from the abused server's location to the source of the forged packets without any administrative cooperation. This provides similar capabilities to the translation feature of the next scheme.

Clean Slate Approaches

Instead of trying to work around the limitations of IP, Accountable Internet Protocol (AIP) [ABF+08] takes a more drastic approach and does away with IP addresses as well as the existing CIDR network delegation system, and replaces these with a hierarchy of *self-certifying* identifiers [MKKW99]. These new cryptographic addresses allow routers to refuse to forward packets until the claimed sender has verified that they generated the packet, much like 802.1x does for switches at the link layer, but without a global cryptographic infrastructure. AIP goes well beyond just preventing forged source addresses, and also addresses problems such as host mobility and secure interdomain routing.

With AIP, the Internet continues to be structured as a collection of autonomous systems (ASs). Each AS contains one or more *accountability domain* (AD) which roughly correspond to advertised BGP prefixes in today's network architecture. Hosts are identified by an *endpoint identifier* (EID), which is globally unique but arbitrary (it is the hash of the host's current public key). Full addresses in AIP have the form `AD:EID`, with interdomain routing using only the `AD` portion and final delivery within the AD using the `EID`.

Unlike IP addresses, ADs and EIDs contain no internal structure related to network topology. In fact, each is simply the hash of the public key for that AD or EID, hence the self-certifying property. Instead of publishing the binding between address and public key as you would when using a public key infrastructure, in a

self-certifying system the public key *is* the address.[2] While these keys could be used, for example, with IPSec [KS05], their primary purpose is securing the network layer and not providing end-to-end security. An effect of only focusing on sender authentication is that each AIP packet does not need to be signed (multiple times), and routers do not perform per-packet cryptographic operations. Instead, the first time an EID or AD forwards a packet to a router, a challenge is issued requiring the sender reply with proof that they possess the private key associated with the public key represented by the EID or AD. Periodically already recognized senders are forced to re-authenticate in order to continue sending packets.

The protocol to verify a source address within its own AD is as follows: When a packet with source *S* is received the *accept cache* at router *R* is checked, and if *S* is present the packet is accepted. If *S* is not present in the cache, the packet is dropped and a verification message is sent to *S* containing the source and destination addresses of the packet, a hash of the packet, and the interface on *R* which the packet arrived on, along with a HMAC of these four values using a secret key known only to *R*. When a host receives this verification message, it consults a local cache of recently sent packets to determine if it did indeed send the packet in question. If so, it replies to *R* with a message consisting of 'accept', its public key, and the entire verification message, all signed with its own private key. It then resends the packet which triggered verification, since *R* does not queue packets while waiting for verification.

Interdomain verification of a packet from *S* as it crosses from AD A to AD B can occur in one of three ways: first, if A and B mutually trust one another to properly verify senders, as would likely be the case for tier-1 backbone ADs, the packet will be forwarded automatically. Otherwise, B uses unicast Reverse Path Forwarding (uRPF) to verify that the route to *S* from B uses the interface which the packet arrived on. This check passes when A is a single-homed network and B is near the edge of the Internet. Finally, if neither of these tests pass, B verifies *S* as if the source's AD were B, and caches the result. To limit thrashing in the accept cache, when the number of individual EIDs for a single AD crosses some threshold *T*, the individual entries are replaced with a wildcard accept for and EID from that AD.

As a result of this online authentication process before forwarding packets, impersonation attacks are limited to hosts within the same AD as the forged source, and, depending on the topology, the same subnet or even switch as the forged source[3]. Anonymity, or "address minting" as it's termed in [ABF+08], becomes easier within the attacker's home AD, however, and if the attacker has control of an AD, they can fabricate new, fake ADs and addresses within them. Because EIDs are essentially random and chosen by the host not the network, AID does not have a way to restrict hosts from generating new EIDs or rapidly changing EIDs to avoid filters or traces. To defend against these attacks, the authors propose utilizing rate-limiting on router interfaces or switch ports to prevent rapid creation of new EIDs.

A weakness of the cryptographic analysis in [ABF+08] is that they assume only individual hosts or stub ADs will be malicious, and that no ADs are misconfigured. If either of these assumptions does not hold, identifying the origin of packets may be as challenging in AIP as it is in IP. Because of the way packets are accepted or verified between ADs, only the first few hops are likely to be actively verified. The core of the network will implicitly trust neighboring ADs, while the ADs nearest the destination will pass uRPF unless asymmetric routing occurs. This reliance on catching forged packets early makes the security guarantees of AIP a "crunchy shell around a soft, chewy center".

The deployment challenges for AIP are significant, requiring not just 100% deployment on routers but also support in the networking stacks of all end hosts. No partial or bridged deployment approach (such as IPv6-on-IPv4 overlays) is proposed, so the practicality of AIP is almost zero. They do show, however, that

[2] An obvious ramification of this is that raw addresses are no longer easily memorable and that two adjacent ADs or EIDs will have completely different addresses. AIP assumes a secure DNS exists for hostname to address lookups.

[3] The authors propose pushing AIP verification into Layer 2, authenticating individual switch ports

technologically there are no significant performance issues caused by switching from routing based on structured prefixes to flat labels. Their analysis of the overhead cryptography has on performance is notably flawed: they argue gains in CPU performance will overcome currently unacceptable costs of RSA signature generation/verification, not scaling keysizes to counter the decrease in security these same CPU performance increases cause.

A second significant hurdle to deployment which is outside the scope of AIP itself is the problem of key-management. The problem of consistent, memorable naming was alluded to previously, and assumes secure authenticated DNS. ADs must learn the identities of their peers as well, which the authors suggest be done offline. The authors also examine key compromise, particularly the issue of detecting an attacker impersonating a host or AD through a compromised private key. The solution proposed to this is to maintain a global registry of identities, revocations, peerings, and AD membership. Each host and AD is responsible for periodically verifying the data in this registry and detecting impostors. Some of these lists, such as the mapping between EID and ADs it belongs to scale with the number of active hosts on the Internet!

Bibliography

[ABF+08] David G. Andersen, Hari Balakrishnan, Nick Feamster, Teemu Koponen, Daekyeong Moon, and Scott Shenker. Acountable Internet protocol (AIP). In *Proc. of ACM SIGCOMM '08*, August 2008. 41, 42

[Acc] Accuris Networks. Accuguard lawful intercept solution [online]. http://www.accuris-networks.com/assets/files/pdf/lawful/AccuGUARD_Datasheet.pdf [accessed 9 July, 2010]. 15

[ACF01] Charles J. Antonelli, Kevin W. Coffman, and J. Bruce Fields. The 10Mbps advanced packet vault. CITI Technical Report 01-10, The University of Michigan, October 2001. 37

[ANSI03] American National Standards Institute. Lawfully authorized electronic surveillance. ANSI Joint Standard J-STD-025B, TIA/ATIS, August 2003. 10

[ANSI07] American National Standards Institute. Lawfully authorized electronic surveillance (LAES) for Internet access and services. ANSI Standard ATIS-1000013.2007 and ATIS-1000013.a.2009, ATIS, March 2007. 11

[Aqs] Aqsacom. AQSACOM lawful interception system (ALIS) platform [online]. http://www.aqsacomna.com/us/index.cfm?vSectionCode=PROD1 [accessed 9 July, 2010]. 15

[AUH99] Charles J. Antonelli, Matthew Undy, and Peter Honeyman. The packet vault: Secure storage of network data. In *Proc. of 1st Workshop on Intrusion Detection and Network Monitoring (ID '99)*. The USENIX Association, April 1999. 37

[Axe00] Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. In Ravi Sandhu, John McLean, and Eugene Spafford, editors, *Transactions on Information and System Security*, volume 3(3), pages 186–205. ACM Press, August 2000. 36

[BBF+00] Steven M. Bellovin, Matt Blaze, David Farber, Peter Neumann, and Eugene Spafford. Comments on the Carnivore system technical review, December 2000. 13

[BC00] Hal Burch and Bill Cheswick. Tracing anonymous packets to their approximate source. In *Proc. of 14th LISA*, page 319–327. USENIX Association, December 2000. 39, 41

[BFS04] Fred Baker, Bill Foster, and Chip Sharp. Cisco architecture for lawful intercept in IP networks. RFC 3924, Internet Engineering Task Force, October 2004. 11

[Blo70] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970. 38

[BPA04] Philip Branch, Ana Pavlicic, and Grenville Armitage. Using MAC addresses in the lawful interception of IP traffic. In *Australian Telecommunications Networks & Applications Conference 2004*, Sydney, Australia, December 2004. 38

[Cact] The Cacti Group. Cacti: the complete rrdtool-based graphing solution [online]. `http://www.cacti.net/` [accessed 7 May, 2010]. 37

[CAL01] CALEA Implementation Section. Flexible deployment assistance guide: Packet-mode communication. OMB Control Number 1110-0030, Department of Justice/Federal Bureau of Investigation, August 2001. Second Edition. 6

[Cas04] Eoghan Casey. *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*. Elsevier Academic Press, London; San Diego, Calif., second edition, 2004. 37

[Cis07] Cisco Systems, Inc. Catalyst switched port analyzer (SPAN) configuration example. Document ID 10570, Cisco Systems, July 2007. 11

[Cla05] Richard Clayton. *Anonymity and Traceability in Cyberspace*. PhD thesis, University of Cambridge, Darwin College, 2005. 38

[Com09] Computer Crime and Intellectual Property Section. *Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations*. Criminal Division, United States Department of Justice, 3rd edition, September 2009. 37

[CSB06] Eric Cronin, Micah Sherr, and Matt Blaze. On the reliability of current generation network eavesdropping tools. In Martin S. Oliver and Sujeet Shenoi, editors, *Advances in Digital Forensics II: Proc. of 2nd Annual IFIP WG 11.9 International Conference*, volume 222 of *IFIP International Federation for Information Processing*, pages 199–214. Boston: Springer, 2006. 26

[CSB08] Eric Cronin, Micah Sherr, and Matt Blaze. On the (un)reliability of eavesdropping. In Nasir Memon and Rajni Goel, editors, *Special Issue on Network Forensics*, volume 3(2) of *International Journal of Security and Networks*, pages 103–113. Inderscience, 2008. 26

[CSVD04] Benoit Claise, Ganesh Sadasivan, Vamsi Valluri, and Martin Djernaes. Cisco Systems NetFlow services export version 9. RFC 3954, Internet Engineering Task Force, October 2004. 13, 37

[CTL09] Cable Television Laboratories, Inc. Cable broadband intercept specification. Data-Over-Cable Service Interface Specifications CM-SP-CBI2.0-I03-090121, CableLabs, January 2009. 6, 11

[DCW+99] Robert Durst, Terrence Champion, Brian Witten, Eric Miller, and Luigi Spagnuolo. Testing and evaluating computer intrusion detection systems. *Communications of the ACM*, 42(7):53–61, July 1999. 36

[Den87] Dorothy E. Denning. An intrusion-detection model. In Virgil D. Gligor and David J. Bailey, editors, *Transactions on Software Engineering*, volume 13(2), pages 222–232. IEEE Computer Society, February 1987. 36

[Der06] DerEngel. *Hacking the Cable Modem: What Cable Companies Don't Want You to Know*. No Starch Press, September 2006. 30

[Der08] Luca Deri. nCap high speed packet capture and transmission library [online]. December 2008. http://luca.ntop.org/nCap/ [accessed 13 July, 2010]. 11

[Der10] Luca Deri. Port mirror vs network tap [online]. January 2010. http://www.ntop.org/blog/?p=14 [accessed 13 July, 2010]. 11

[EFF06] Electronic Frontier Foundation. FOIA litigation: Electronic surveillance systems [online]. October 2006. http://www.eff.org/issues/foia/061708CKK [accessed 7 May, 2010]. 13

[EPIC05] Electronic Privacy Information Center (EPIC). Carnivore FOIA documents [online]. January 2005. http://www.epic.org/privacy/carnivore/foia_documents.html [accessed 7 May, 2010]. 13

[ETI] ETI Connect. Lawful intercept network connector (LINC) [online]. https://www.eticonnect.net/index.php?id=52 [accessed July 9, 2010]. 15

[ETSI07] European Telecommunications Standards Institute. Lawful interception (LI); Handover interface for the lawful interception of telecommunications traffic. ETSI Standard ES 201 671 V3.1.1, ETSI, May 2007. 38

[Ette] Ettercap NG [online]. http://ettercap.sourceforge.net/ [accessed 12 July, 2010]. 13, 50

[FCC05] Federal Communications Commission. First report and order and further notice of proposed rulemaking in the matter of Communications Assistance for Law Enforcement Act and broadband access and services. FCC Docket No. 05-153, Federal Communications Commission, September 2005. 4, 11

[FCC06] Federal Communications Commission. Second report and order and memorandum opinion and order in the matter of Communications Assistance for Law Enforcement Act and broadband access and services. FCC Docket No. 06-56, Federal Communications Commission, May 2006. 4

[Flu] Fluke Networks. Clearsight analyzer [online]. http://www.flukenetworks.com/fnet/en-us/products/ClearSight+Analyzer/ [accessed 9 July, 2010]. 15

[GFD+05] Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Škorić. An information-theoretic measure of intrusion detection capability. Technical Report GIT-CC-05-10, College of Computing, Georgia Institute of Technology, 2005. 36

[GFD+06] Guofei Gu, Prahlad Fogla, David Dagon, Wenke Lee, and Boris Škorić. Measuring intrusion detection capability: An information-theoretic approach. In *Proc. of 2006 ACM Symposium on Information, computer and communications security (ASIACCS '06)*, pages 90–101. ACM Press, March 2006. 36

[gig] Gigamon [online]. http://www.gigamon.com/ [accessed 9 July, 2010]. 11

[GU01] John E. Gaffney, Jr. and Jacob W. Ulvila. Evaluation of intrusion detectors: A decision theory approach. In Roger Needham and Martin Abadi, editors, *Proc. of 2001 IEEE Symposium on Security and Privacy*, pages 50–61. IEEE Computer Society, May 2001. 36

[Har05] Nick Harbour. tcpxtract [online]. October 2005. http://tcpxtract.sourceforge.net/ [accessed 7 August, 2010]. 12

[HBP06] Mark Huang, Andy Bavier, and Larry Peterson. PlanetFlow: Maintaining accountability for network services. *Operating Systems Review*, 40(1):89–94, January 2006. 37

[HPK01] Mark Handley, Vern Paxson, and Christian Kreibich. Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics. In *Proc. of 10th USENIX Security Symposium*. The USENIX Association, August 2001. 24, 36

[IEEE90] IEEE Computer Society. Token-passing bus access method and physical layer specifications. IEEE Std. 802.4, LAN/MAN Standards Committee, August 1990. 27

[IEEE03] IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Std. 802.11, LAN/MAN Standards Committee, June 2003. 27

[IEEE05] IEEE Computer Society. Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. IEEE Std. 802.3, LAN/MAN Standards Committee, December 2005. 27

[IP] IP Fabrics. DeepSweep for CALEA [online]. http://www.ipfabrics.com/products/DeepSweep-CALEA.php [accessed July 9, 2010]. 15

[ipo] ipoque. DPX network probe [online]. http://www.ipoque.com/products/dpx_network_probe. 15

[IS03] Alex Iliev and Sean Smith. Prototyping an armored data vault: Rights management on big brother's computer. In *2nd International Workshop on Privacy Enhancing Technologies. Revised Papers (PET 2002)*, volume 2482 of *Lecture Notes in Computer Science*, pages 234–238. Springer, 2003. 37

[Joh06] Thomas A. Johnson, editor. *Forensic Computer Crime Investigation*. Forensic Science Series. CRC Press Taylor & Francis Group, 2006. 37

[KKKM06] Panagiotis Kanellis, Evangelos Kiountouzis, Nicholas Kolokotronis, and Drakoulis Martakos, editors. *Digital Crime and Forensic Science in Cyberspace*. Idea Group Publishing, 2006. 37

[KPD+05] Stefan Kornexl, Vern Paxson, Holger Dreger, Anja Feldmann, and Robin Sommer. Building a time machine for efficient recording and retrieval of high-volume network traffic. In *Proc. of 2005 Internet Measurement Conference*, pages 267–272. The USENIX Association, October 2005. 37

[Kre] Christian Kreibich. The NETwork DUmp data Displayer and Editor [online]. http://netdude.sourceforge.net/ [accessed 7 May, 2010]. 33

[KS05] Stephen Kent and Karen Seo. Security architecture for the Internet Protocol. RFC 4301, Internet Engineering Task Force, December 2005. (obsoletes IETF RFC 2401). 41, 42

[LFG+00] Richard P. Lippmann, David J. Fried, Isaac Graf, Joshua W. Haines, Kristopher R. Kendall, David McClung, Dan Weber, Seth E. Webster, Dan Wyschogrod, Robert K. Cunningham, and

Marc A. Zissman. Evaluating intrusion detection systems: The 1998 DARPA off-lineintrusion detection evaluation. In *Proc. of 2000 DARPA Information Survivability Conference and Exposition (DISCEX '00)*, volume 2, pages 12–26. IEEE Computer Society, January 2000. 36

[LX01] Wenke Lee and Dong Xiang. Information-theoretic measures for anomaly detection. In Roger Needham and Martin Abadi, editors, *Proc. of 2001 IEEE Symposium on Security and Privacy*, pages 130–143. IEEE Computer Society, May 2001. 36

[Mer09] Merit Network Inc. OpenCALEA [online]. 2009. http://www.opencalea.org/ [accessed 9 Jul, 2010]. 13

[MKKW99] D. Mazieres, M. Kaminsky, M. F. Kaashoek, and E. Witchel. Separating key management from file system security. In *Proc. of 17th ACM SOSP*, page 124–139, December 1999. 41

[Neta] Net Optics, Inc. Net Optics Director [online]. http://www.netoptics.com/products/product_family.asp?cid=9&Section=products&menuitem=9 [accessed 9 July, 2010]. 11

[Netb] NetScout Systems, Inc. nGenius InfiniStream deep packet capture and analysis appliances [online]. http://www.netscout.com/products/infinistream.asp [accessed 13 July, 2010]. 11

[Netc] NetWitness. NetWitness Investigator [online]. http://www.netwitness.com/products/investigator.aspx [accessed July 9, 2010]. 15

[Netd] NetworkMiner. NetworkMiner forensic analysis tool (NFAT) [online]. http://networkminer.sourceforge.net/ [accessed July 9, 2010]. 15

[Net06] NetOptics, Inc. *Installation Guide for 10 GigaBit Fiber Slim Tap*, October 2006. Rev.2. 11

[Net07] NetOptics, Inc. CALEA compliance: Utilizing tap technology to meet assistance capability requirements. Executive brief, NetOptics, Inc., 2007. 11

[Net09] NetOptics, Inc. Use of taps and span ports in cyber intelligence applications. White paper, NetOptics, Inc., September 2009. 11

[ngre] ngrep – network grep [online]. http://ngrep.sourceforge.net/ [accessed 12 July, 2010]. 13

[NTOP] NTOP – Network TOP [online]. http://www.ntop.org [accessed 7 May, 2010]. 37

[Pax99] Vern Paxson. Bro: A system for detecting network intruders in real-time. *Computer Networks*, 31(23-24):2435–2463, December 1999. 36

[PN98] Thomas Ptacek and Tim Newsham. Insertion, evasion, and denial of service: Eluding network intrusion detection. Technical report, Secure Networks, Inc., Calgary, Alberta, Canada, January 1998. 24, 28, 29, 36

[Pos81] Jon B. Postel. Internet Protocol. RFC 791, Internet Engineering Task Force, September 1981. 25, 26, 28, 39

[QoS] QoSient, LLC. The network Audit Record Generation and Utilization System (Argus) [online]. http://www.qosient.com/argus/ [accessed 7 May, 2010]. 13, 37

[RVC01] Eric C. Rosen, Arun Viswanathan, and Ross Callon. Multiprotocol label switching architecture. RFC 3031, Internet Engineering Task Force, January 2001. 25

[Sag98] Glenn Sager. Security fun with OCxmon and cflowd. In *Internet-2 Measurement Working Group Meeting*, November 1998. Slides available at http://www.caida.org/funding/ngi1998/content/security/1198/. 38

[SCCB05] Micah Sherr, Eric Cronin, Sandy Clark, and Matt Blaze. Signaling vulnerabilities in wiretapping systems. *IEEE Security & Privacy Magazine*, 3(6):13–25, November/December 2005. 10, 18, 21, 30

[SHHPK+00] Stephen P. Smith, Jr. Henry H. Perritt, Harold Krent, Stephen Mencik, J. Allen Crider, Mengfen Shyong, and Larry L. Reynolds. Independent review of the Carnivore system. Final Report IITRI CR-030-216, IIT Research Institute, December 2000. 6, 13

[SM96] Tsutomu Shimomura and John Markoff. *Takedown: The Pursuit and Capture of Kevin Mitnick, America's Most Wanted Computer Outlaw-By the Man Who Did It*. Hyperion Books, 1996. 37

[SMSB03] Kulesh Shanmugasundaram, Nasir Memon, Anubhav Savant, and Herve Bronnimann. ForNet: A distributed forensics network. In Vladimir Gorodetsky, Leonard Popyack, and Victor Skormin, editors, *Proc. of 2nd International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS '03)*, volume 2776 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag Berlin / Heidelberg, September 2003. 37

[Sol] Solera Networks. DeepSee [online]. http://www.soleranetworks.com/products/network-forensics-software. 15

[Son] Dug Song. Fragroute: Intercept, modify, and rewrite egress traffic [online]. http://monkey.org/~dugsong/fragroute/ [accessed 7 May, 2010]. 29

[SP01] Dawn Song and Adrian Perrig. Advanced and authenticated marking schemes for IP traceback. In *Proc. IEEE INFOCOM*, volume 2, page 878–886, April 2001. 41

[SP03] Umesh Shankar and Vern Paxson. Active mapping: resisting NIDS evasion without altering traffic. In Steven M. Bellovin and David A. Wagner, editors, *Proc. of 2003 IEEE Symposium on Security and Privacy*, pages 44–61. IEEE Computer Society, May 2003. 36

[Spa89] Eugene H. Spafford. The Internet worm: Crisis and aftermath. *Communications of the ACM*, 32(6):678–687, June 1989. 37

[SPS+02] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent, and W. Timothy Strayer. Single-packet IP traceback. *IEEE/ACM Transactions on Networking*, 10(6):721–734, December 2002. 38, 39

[Sto89] Clifford Stoll. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. Doubleday, 1989. 37

[SWKA01] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Network support for IP traceback. *IEEE/ACM Transactions on Networking*, 9(3), June 2001. 38, 39, 40, 41

[TCP] Tcpdump/libpcap [online]. http://www.tcpdump.org/ [accessed 7 May, 2010]. 11

[Tec01] Technical Working Group for Electronic Crime Scene Investigation. 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, July 2001. 37

[Tec04] Technical Working Group for the Examination of Digital Evidence. 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, April 2004. 37

[TIA02] TIA TR45 Lawfully Authorized Electronic Surveillance Ad Hoc Group. Summary of CALEA requirements, November 2002. Version 2.1 (available at http://cryptome.org/laes/calea-require.pdf). 4

[Top] TopLayer Security. Data collection filtering device (DCFD) [online]. http://www.toplayer.com/content/products/others/dcfd.jsp [accessed July 9, 2010]. 15

[Uni04] United States Department of Justice, Federal Bureau of Investigation, and Drug Enforcement Administration. Joint petition for rulemaking to resolve various outstanding issues concerning the implementation of the communications assistance for law enforcement act. Joint petition for expedited rulemaking, Federal Communications Commission, March 2004. 10

[USC68] United States Congress. Omnibus crime control and safe streets act of 1968. Pub. Law No. 90-351, 82 Stat. 197, United States of America, June 1968. 4

[USC78] United States Congress. Foreign intelligence surveillance act of 1978. Pub. Law No. 95-511, 92 Stat., United States of America, 1978. 4

[USC94] United States Congress. Communications assistance for law enforcement act. Pub. Law No. 103-414, 108 Stat. 4279, United States of America, October 1994. 4, 10

[Wil] WildPackets. OmniPeek [online]. http://www.wildpackets.com/products/network_analysis_and_monitoring/omnipeek_network_analyzer [accessed July 9, 2010]. 15

[Wire] Wireshark network protocol analyzer [online]. http://www.wireshark.org/ [accessed 7 May, 2010]. 13

Image Credits

Figure 2.1: From [Ette], http://ettercap.sourceforge.net/screenshots.php

Figure 2.2: From [Ette], http://ettercap.sourceforge.net/screenshots.php