Proceedings of the 2001 IEEE
Workshop on Information Assurance and Security
United States Military Academy, West Point, NY, 5-6 June, 2001

Tracing the Source of Network Attack: A Technical, Legal and Societal Problem

S. C. Lee* and C. Shields**

*The Johns Hopkins Applied Physics Laboratory,
Laurel, MD 20723, sue.lee@jhuapl.edu
**Purdue University/CERIAS, West Lafayette, IN 47907, clay@cerias.purdue.edu

ABSTRACT

In the field of network defense, very little research is directed toward locating the source of network attacks. This paper models the technical aspects of attack traceback. By analyzing the model of attack traceback, two fundamental technical problems can be identified: determining the immediate source of packets (which may be disguised through IP spoofing) and determining causality for packets arriving at and issuing from a host. The past and on-going research efforts that either directly address or are applicable to attack traceback are tied to the traceback technical model, to show the extent to which they address the problem, and to highlight research gaps. The paper also discusses some of the legal and societal roadblocks to a technical solution

Keywords: Traceback, IP Spoofing, Network Forensics

INTRODUCTION

In the field of network defense, a large body of work addresses myriad aspects of attack prevention, detection and reaction. One preventive technique that is seldom addressed is deterrence. Most attackers are risk adverse. A high rate of discovery and punishment would discourage many would-be attackers; unfortunately, today it is easy to carry out attacks anonymously. Similar problems exist with many types of computer fraud and, in an information warfare scenario, locating the source of a network attack provides information about attackers' identity, physical location, and capabilities.

To date, there has been very little work that is intended to locate the source of network attacks. While there is some past work that attempted to identify the source of interactive streams [1], most recent work addresses the problem of locating the source of packets used for a denial-of-service attack [2, 3, 4] - not with the intention of holding the attacker accountable, but instead with the intention of halting the attack.

This paper explores the technical, social, and legal problems faced by designers of *traceback* systems that attempt to locate the particular host in the network that is initiating network attacks, probes, or computer fraud. By providing an overview of the problem, we hope to encourage active research in the area. Ideally, a traceback system would identify the human agent responsible for the attack, however identification and authentication of a user to a machine will not be addressed in this discussion of traceback, since it is a generic problem pervading information assurance. Traceback is therefore limited to determining the host that is the source of an attack. Sometimes, the identification of the attacker host will be sufficient to strongly implicate a particular human agent. At other times, the attacker host may be so publicly available as to make identification of the actual human agent very difficult. Traditional methods of crime investigation or intelligence gathering can be applied to assign responsibility to some individual.

PROBLEM SPACE

Attack Traceback Model

Every active, network-based attack begins with the issuance of attack packets from an *attacker host*, used as the entry point into the computer network across which the attack occurs. In some cases, a human directly uses the attacker host to launch an attack. In other cases, it will be the point of entry into the computer network from some other communications network, such as a host reached by telephone dial-up. The attacker host generates a series of packets that eventually cause the arrival of attack packets at a *victim host*, the final point on the path through the network that is affected by the attack. The term "host" (generally, a human-useable system such as a PC or workstation) will be used in this paper, although in some cases the attacker or victim may be a network device that is usually used to support packet forwarding. This causes no real loss of generality.

The attack packets seen at the victim contain, in the worst case, a single clue to the identity of the attacker: the source IP address [5]. Most attackers take advantages of one or more techniques to ensure that their identity cannot be learned through the packet source address [2]. A network traceback system aims to determine this address.

One technique for hiding the true source of packets is to simply forge the source address in the transmitted packets. Because IP routing depends on destination address alone, the source IP address in a packet has no necessary

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

connection with the actual source of the packet. Forging an address in a one-way communication is as simple as putting any desired address in the source address field. Forging the source of a two-way communication is more difficult because the forger may not see the traffic passing from the victim to the machine whose addressed is being forged. In attacks that exploit trust relationships based on IP addresses, for example, the attacker must guess the TCP sequence numbers in the response packets. This attack is possible, however, because many operating system implementations use easily guessable sequence number choices in network communication. For a simple, predictable two-communication, the attacker can simply carry out its part of the communication "in the blind". If the attacker lies on the path between two communicating parties, it is also possible to conduct a "session hijacking" attack that takes over a connection established by a legitimate user after authentication has occurred.

The second technique for hiding the source host is by "laundering" the attacker's packets through some intermediate host [6]. This technique involves some application-level mechanism to transform the type or timing of information that leaves the laundering host. Schematically, laundering can be depicted as:

p1(*src:* attacker, *dest:* launderer, contents, t) →
p2(*src:*launderer, *dest:* victim, T(contents), t+δt)

A packet (p1) is issued at time t from the attacker host. It has the source address of the attacker host, destination address of the laundering host, and some contents. Reception of p1 causes the laundering host to issue its own packet (p2). P2 contain the source address of the laundering host. This change alone disguises the attacker host's identity. In addition, the contents of the attacker host's packet may undergo some transformation T. Finally, issuance of the laundering host's packet may follow reception of the attacker host's packets by some time δt. The magnitude of the transformation T and the timing delay δt may be trivial or profound. In fact, the only connecting link between p1 and p2 may be one of causality. That is, the arrival of the attacker host's packets in some way causes the laundering host to send its packets. While communications between the attacker, laundering and victim hosts may be two-way, the return communications are not disguised.

To illustrate the transformation of contents, suppose that the attacker host telnets to a laundering host, and then opens a secure shell for communication to the victim host. While the attacker is opening the shell, his commands to the laundering host are executed on that host. These commands cause the laundering host to initiate a connection to the victim. During this stage of the attack, the laundering host transforms the attack packet contents from Unix commands (arriving at the launderer) to TCP connection establishment (leaving the launderer). Once the connection is established, however, the attacker's packets contain commands intended for the victim host. The laundering host no longer executes (transforms) them; it simply repackages them and passes them on.

A timing delay may be created innocently through system processing delays, or be deliberately introduced by the attacker to disguise his role in the attack. This could be done either to put time between his interaction with the laundering host and the laundering host's interaction with the victim, or to disguise the timing signature of his real time communications with the victim through the laundering host. For example, an attacker may introduce a script on a laundering host, set to run at a future date. Alternately, he could install a program on the laundering host to introduce delays in the passage of his attack packets.

Figure 1 depicts a simple model of an attack that incorporates all the elements an attacker might use to hide his identity, although not in every possible combination. Each box on this diagram represents one (or more) hosts on the attack path. These hosts represent the typical ways that attackers use laundering hosts to disguise their identity. A single attack may not incorporate all different types of hosts and obscuration techniques.

The attacker host may initiate communications with a stepping stone host - a compromised host that acts as a conduit for the attacker host's communications. By definition, the attacker's communications are not fundamentally transformed or delayed by a Stepping Stone host. Although the communications flowing through the stepping stone are unchanged in *essence*, they may be changed in superficial but confounding ways. For example, content may appear to change because of encryption. Random timing changes may occur as the communications pass in and out. An attack may pass through more than one stepping stone host before reaching the victim. The classic penetration attack is usually conducted through multiple stepping stone hosts to prevent identification of the attacker.

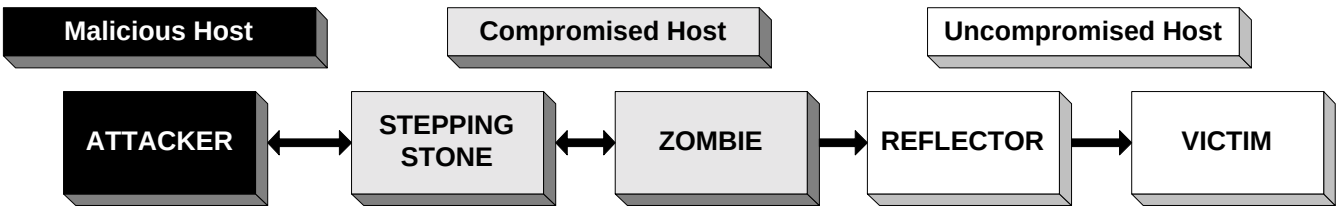

Figure 1 – Attack Traceback Model

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

The attacker host may initiate communications with a zombie host, either directly or through one or more stepping stone hosts. By definition, a zombie is a laundering host that fundamentally transforms and/or delays the attacker's communications before they continue down the attack path. For example, the attacker host may install a Trojan on the zombie host timed to execute minutes, days or even weeks after the attacker's contact. Thus the attacker host's communications may not be contiguous in time with the downstream attack communications. The content of the communications may be transformed as well. For example, a single command input from the attacker host may trigger execution of a planted script that issues a series of entirely different commands down the attack path. The distributed denial-of-service attack uses zombie hosts in this way [7].

The next host in the traceback model depicted in Figure 1 is a reflector host. Unlike the stepping stone or zombie, the reflector is an uncompromised host that cooperates with the attack in an innocent manner consistent with its normal function. For example, the attacker host (either directly or via stepping stones and/or zombies) may send a packet designed to elicit a response from the reflector. If the victim's source IP is spoofed as the packet source, then the reflector will innocently direct its response toward the victim. The response packet (or packets) constitutes the attack. Because the reflector host does not need to be compromised to participate in the attack, the attacker may never have communicated with the reflector prior to the attack. No trace of the attacker may be present in old log files, as will be the case with any compromised laundering host.

Fundamental Traceback Problems

Analysis of the traceback model shows that there are two distinct sub-problems that are fundamental to the general problem of tracing an attack back to the attacker host. The first is identifying the source of packets. Because of the possibility of IP source address forging, the apparent source of the packets arriving at any downstream host in the traceback model may be incorrect. The first problem in traceback, then, is to learn the true identity of the upstream host in all the host-to-host communication paths.

Once the source of a packet stream is identified as a laundering host, the second problem arises. That problem is one of discovering causality in communications into and out of the same host. The stepping stone host represents a special case of determining causality. Since the stepping stone acts as a mere conduit for the attack, the packets it sends have been sent to it from some host further upstream on the attack path. Potentially, the upstream host can be identified by matching the two communication streams - into and out of the stepping stone. This matching may be complicated by apparent changes in content, for example through encryption, or timing variations caused by natural system delays. To further complicate the problem, the stream matching may not take place on either side of a single host. If the attack path passes through hosts outside the area that can be observed by the traceback, then the stream matching may have to be done between streams into and out of two hosts separated by an unknown number of intermediate stepping stones. In this latter case, not only do the streams need to be matched, but also in the general case, the matching stream may need to found. In general then, traceback through stepping stone hosts requires determining if two communications streams, viewed at different points in the network, have the same origin and are essentially the same stream.

The zombie host represents the far more general causality problem. Here, the upstream communication that resulted in the attack is not similar in content and/or connected in time to the communications downstream from the zombie. All that is known is that, at some point in time, a communication into the zombie caused the observed out-going packet stream. Some intermediate transforming event (e.g., expiration of a timer, execution of a Trojan program) occurs between the in-bound and out-bound portions of the attack.

Traceback Characterization

The result of an attack traceback can be characterized by three parameters: precision, accuracy, and timeliness. Alternative solutions to the attack traceback problem may yield results that differ in these characteristics.

Precision is a measure of the exclusivity of the traceback result. The attacker host might be identified as one of a group of hosts. For example, a traceback might be able to identify the source of an attack to within a particular LAN, to a host connected via one particular ISP, or even as a host in some particular country. Some methods of traceback might even yield a group of hosts that have no particular common tie except that they are all consistent with the clues followed by the traceback. Some methods may yield results of varying precision, depending on the traceback environment encountered during each particular attempt.

Accuracy is a measure of correctness of a traceback; that is, given that a traceback results in identification, how likely is that identification to be correct? Some traceback methods may identify a host as consistent with a particular attack, but not necessarily the only possible attacking host. If a traceback solution can result in a false identification, then ideally the solution will also quantify the probability that the result is correct.

Precision and accuracy may also be a function of how far back on the attack path the traceback goes. For example, some solutions may be able to identify the first in a series of stepping stone hosts with 100% accuracy and precision. Beyond that point, precision, accuracy or both may degrade.

Timeliness is a measure of when a traceback result can be obtained. Some solutions may only give a result while an

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

attack is in progress. Other solutions may require access to data that can only be acquired post-attack. Some solutions may depend on the existence of data that has a limited lifetime. Versatile traceback solutions, usable during, immediately following, or long after an attack, are probably less tractable than solutions that operate in a specific timeframe.

Environmental Factors

A number of factors can either simplify or complicate the execution of an attack traceback. One of these is the network environment. The network environment for traceback ranges from completely controlled to totally uncontrolled. The controlled environment is a network under a single administration that dictates the network and desktop configurations. Tools for traceback (e.g., altered routers, specific host or network monitoring) can be mandated, and the potential sources and types of network communications are constrained. An example of traceback in a totally controlled environment might be finding the source of an insider attack within a highly secure network. In this environment, the traceback problems are more tractable. Typically, the network environment will be less controlled. For example, the typical intranet where the administrators control the network but not the desktop configurations is a partially controlled environment. The Internet today typifies an entirely uncontrolled environment. It is possible that the general traceback problem cannot be solved in the totally uncontrolled environment. This possibility leads to a host of research questions regarding the degree of control over the environment that is necessary for traceback to succeed, at least some substantial portion of the time.

Another factor that affects traceback is the nature of the attack itself. The salient attack features are the amount of traffic and time extent of the attack. At one extreme, an attack might consist of a single packet originating at the attacker host. The general traceback problem might not solvable for this case; on the other hand, this extreme may not be representative of any real attacks. In general, an attack generates a certain amount of traffic along the attack path, and is on-going for some time extent. The more information that is transferred and the longer that the attack is in progress, the more tractable the traceback problem becomes. If the usual preparatory period is included in the definition of the "attack", then the typical attack may involve a substantial amount of traffic and time extent. On the other hand, the preparatory period may pass unnoticed until the actual attack occurs. If so, not all traceback solutions can take advantage of it.

Traceback Applications

There are a number of applications for traceback. Each of these has unique requirements in terms of traceback accuracy, precision, and timeliness. Some of the possible applications for traceback are attack reaction (that is, stopping an on-going attack), future attack prevention, establishment of liability, and prosecution of the attacker.

For attack reaction, a traceback solution must operate in real time, and have good accuracy. The accuracy of the identification will, of course, determine how effective the reaction is; if the source of the attack is inaccurately identified, no action taken against that source can stop the attack. High precision may not be required. For example, if an on-going attack can be traced back to some particular input port on a router, a certain LAN, or even a domain, filtering may be used to end the attack. Of course, the precision of the traceback and the subsequent filtering determines how much impact the attack reaction has on nominal traffic.

For future attack prevention, the traceback does not have to operate in real time as long as it can be completed in time to take preventive measures before the next attack. In general, the precision must be much higher than for attack reaction. For example, while it might be acceptable to filter traffic from a fairly large segment of the Internet for a short time to restore partial operation during an attack, such an approach is unacceptable in the long term. If the attack can be identified as coming from some relatively bounded portion of the network, however, more preventative options are available. If a very precise identification can be made, selective filtering and possibly even arrest and prosecution are viable prevention measures. If the attack can be identified as originating from some network under a single administration, the evidence could persuade the administrators to institute stricter security measures. An attack identified as originating from a particular country could elicit diplomatic or military actions. As with attack reaction, the ultimate effectiveness of attack prevention will depend on the accuracy of the identification. The penalty for an inaccurate identification can be more serious than merely remaining open to further attack; for example, a military action against the wrong country would have very far-reaching and unfortunate side effects.

For the purpose of establishing liability, the traceback must occur in a timeframe consistent with the existence of ephemeral data and the statue of limitations that applies to each particular case. High precision may not be required; in fact, it can be argued that traceback to the first entity with "deep pockets" - such as a large corporation intranet or a large ISP - is best for in this case. Criminal prosecution differs from liability in that high precision is required. In general, obtaining a criminal conviction will require identification of one or more individual attackers. A high probability of accuracy may be enough to obtain a favorable verdict in either case.

Consideration of the various applications for traceback leads to several general observations. First, the applications for traceback overlap in some areas. For example, criminal prosecution is one way to prevent a future attack (at least, by that individual). When they do overlap, then the most

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

stringent set of requirements for the traceback apply. Second, good to high accuracy is required in every case. Thus, accuracy considerations should drive the search for a traceback solution. Finally, it should be noted that the application requirements for traceback do not need to be met completely by any automated or computer-aided technical solution. For example, if an automated traceback can identify several candidate sources for an attack, then the candidates may be winnowed down using "outside" information, such as work records, interviews, telephone records, etc.

SOLUTIONS SPACE

Each of the fundamental traceback problems identified above requires a unique solution. Some approaches for each are described below. Integration of the individual problem solutions into an infrastructure will be needed to perform a complete traceback. Pure technical solutions for traceback may be possible; however, the technical solutions cannot be implemented unless they conform to certain legal and societal constraints. Each of these topics is addressed briefly below.

Packet Source Identification

The problem of identifying the source of any given packet arriving at a host is tantamount to tracing the passage of the packet backwards through the switching fabric of the network. Thus the candidate solutions for this fundamental part of a complete traceback rely on detecting which routing devices handled the packet. Given enough resources, this is the one fundamental traceback problem that can be said to have a guaranteed technical solution. Suppose for example, that every routing device in use today could be instantly replaced with one that implements the following process: When receiving a packet from a directly-connected host, the router clears out a "route trace table" (sized to accommodate the maximum number of hops), and places in it the physical address of the directly-connected host and its own unique router ID. A router that receives a packet from another router places its ID into the next available slot in the table. At the destination host, the entire route is contained within each packet. If implemented in the routing device hardware, where it could not be subverted by any network-based attack, then at least the source router for all packets could be uniquely and securely identified. The source host could not be spoofed either, without spoofing the physical address of the host, more difficult than spoofing the source IP in the packet header.

The practical drawbacks to this scheme are immediately obvious: implementation adds to the routing overhead and requires changes to all routing device hardware, the network protocol and the minimum size of a packet. The current research initiatives in route traceback focus on addressing these practical difficulties. Three distinct approaches have been developed: 1) identifying the route within the packet without increasing the packet size, 2) identifying the route using extra packets generated by the routing devices, and 3) actively querying routing devices about traffic they have handled.

The first approach requires overload of some field already present in the IP packet header with route identification material [2]. The essence of this approach is to encode a unique route through a potentially large number of devices within a strictly limited number of bits, and furthermore, to do it in a secure (unspoofable) way. Although there is an obvious limit to the amount of information that can be contained within a limited field, the fact that each router is only connected to a few other routers makes the problem potentially solvable. Even if an absolutely unique route cannot be specified, the number of possible routes may be pruned considerably. One scheme, proposed by Dawn Song and Adrian Perrig of the University of California at Berkeley [3], uses a variation on the Time-Efficient Stream Loss-tolerant Authentication (TESLA) protocol to place a code based on the IP addresses of the routing devices that sequentially handle a packet into the IP identification field. Using a map of the IP addresses of all upstream routers, a destination host can reconstruct the route of a packet through up to 32 devices with reasonable computational efficiency and precision [4].

The second approach requires routing devices to emit a secondary "trace" packet for each packet they handle that is to be traced [8]. At the destination host, both the original packet and all the associated "trace" packets are collected, and a route for the original packet can be reconstructed. The advantage to this scheme is that the trace packet can contain unambiguous, authenticated identification of the originating router. The clear disadvantage is that, if every packet must be traced, an enormous increase in network traffic will ensue. In a scheme of this type proposed by Steve Bellovin, the routing devices emit a trace packet on a probabilistic basis (about 1 in 20,000 for Bellovin's scheme) so that the increase in traffic is minimized, and it is not possible to guess when a trace packet will be issued. It is unlikely that any single packet will trigger enough trace packets to reconstruct a route, however, a large stream of packets continuing for a considerable time (for example, a SYN flood DOS attack stream) can almost certainly be traced this way.

The last approach requires a destination host that desires to find the true source of a packet to send a query to its routing device. This routing device can pass the query upstream to any routing devices to which it is connected. Routing devices responding positively also repeat the query to any upstream routers to which they are connected. Examination of the positive and negative responses yields the packet route. This scheme also generates additional network traffic, however since only suspicious packets are traced, the overhead may not be large. It also requires routing devices to store information on all packets they handle for some period of time. Due to processing and memory limitations, this time period is likely to be quite short, limiting this approach to a near-real time traceback.

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

Finally, this approach requires a query and response protocol that is understood by all the different types of routing devices participating in the traceback. One implementation of a scheme that uses this approach is the Intrusion Detection and Isolation Protocol (IDIP) developed by a Network Associates/Boeing/University of California at Davis/Silicon Defense collaboration sponsored by DARPA [9].

Although any route-tracing scheme benefits by the participation of all routing devices, non-participation by some does not necessarily preclude a successful traceback. The level of participation required for a successful traceback, how the accuracy and precision vary with participation level, and how the traceback approach and/or algorithm affects the required level of participation are all interesting research questions for the traceback community.

Stream Matching

The second fundamental traceback problem is to trace an attack packet stream through some number of stepping stone hosts. Two characteristics have been proposed to match attack streams on either side of laundering hosts: the packet contents and the inter-packet timing. In addition, there are two possible locations for sensing stream characteristics: internal to each host (that is, each host matches streams entering and leaving itself) and external to all hosts (that is, by sniffing network traffic).

In the mid-1990's, Stuart Staniford-Chen developed a content-matching scheme while at the University of California at Davis. Called "Thumbprinting" [2], this scheme divides the stream into discrete time intervals and creates digests of packets within each interval. Two streams are compared by computing the similarity of the stream digests. The probability that the computed similarity could result from two random streams is used to determine a match. Staniford-Chen showed that similar thumbprints are far more probable to represent the same stream than two random streams. Although the scheme works quite well for unencrypted streams, encryption makes stream matching by this method impossible.

In on-going research at Purdue, the second approach to stream matching has shown some early success. Despite random network delays, the characteristic give-and-take of network connection protocols and human-computer interactions generate a "timing thumbprint". The timing thumbprints of a single stream viewed at two points in the network are more similar than the timing thumbprints of unrelated streams. Of course, timing is not affected by encryption, so this method may be more robust than the content-based approach. A second method of stream matching by inter-packet timing has also been suggested. In this approach, the timing of a stream is actively perturbed at some location in the network, and streams at other points in the network are searched for matching perturbations. There may be legal implications to taking this active approach, especially if the stream originates outside of the administrative domain where the active perturbation is applied.

A system that matches streams into and out of a single host has a number of advantages. The host's network stack already sorts packets into coherent streams (connections) so no extra work is required to perform this association. Further, the amount of traffic into and out of a single host is restricted, so that the host can probably perform the needed computations for matching without undue strain. On the other hand, a robust traceback system using host-based matching would require control over all hosts in the attack path. In general, this is unlikely to be the case. A network-based stream-sniffing-and-matching system does not require control over all of the hosts in the attack path, or even a change in the network software on any of them. Any time a matching stream is sniffed at two places in the network, that part of the attack path is known. On the other hand, a network-based system will see a much higher rate of traffic than will any single host. It must sort the traffic into streams, compute the thumbprint on each, and then compute how well each pair of streams sensed at different locations match. It may be quite difficult to keep up with traffic in a network-based scheme. As with route tracing, research into how the number of stream-matching hosts or the number of sniffing points in the network affects the success, accuracy and precision of the traceback is needed.

Additional considerations exist if stream matching is not part of a real time traceback. If an attack traceback is initiated after the attack completes (a common occurrence when the attack is subtle and remains undetected until after the fact), then the stream matching systems must have stored either the thumbprints, or the identifications of matching streams. Due to storage limitations, the timeliness of any traceback requiring stream matching will be affected.

Causality

The final fundamental traceback problem is far and away the most difficult. With neither markings, contents or timing to connect streams into and out of a zombie host, automatically determining that one stream is related to the other seems difficult if not impossible. If any current research addresses this problem, it is not well known in the community.

The bare outlines of a possible solution can be discerned. Determining causality will certainly require access to and examination of the zombie host. The immediate cause of the output stream (e.g., Trojan or script) can be determined. Logs may reveal the trigger (e.g., a remote command or chron job) for the immediate cause. Information about the immediate cause or its trigger can be used to set a window in time to look for the source of the true causal event. This source must be one of the remote connections established within the time window. Clearly, the larger the time window, the more difficult the problem is. Not only will the set of possible sources be larger, but

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

also the retention of logs may limit the ability to create the set. The set of possible sources may be pruned by certain means. For example, a clearly suspicious connection (e.g., from a source unknown to the host users) would merit close attention. Otherwise, each connection must be tracked to its source, and eliminated as suspect or traced further back along the (potential) attack path. Obviously, fundamental thinking is required before any fruitful research can be carried out on this part of the traceback problem.

Traceback Infrastructure

Because the traceback problem has several independent parts, a total solution will require integration of independent solutions into an infrastructure that can invoke the appropriate tools as needed. In addition to integrating tools that address each of the different fundamental aspects of traceback, there will need to be tool variants that work in different timeframes (e.g., during or after an attack). For the foreseeable future, traceback tools will be incomplete; that is, they will require supplemental, non-automated intervention by human agents, particularly to open administrative barriers to traceback, and also to supply supplemental information unavailable to automated processes.

One possible approach to this integration is illustrated by IDIP, mentioned above in the section on packet source identification. The IDIP architecture uses IDIP-enabled devices throughout the network - intrusion detection systems, firewalls, etc. - to implement its query/response protocol for tracing suspect traffic. It uses DARPA's Common Intrusion Detection Framework (CDIF) language to allow these disparate devices to communicate. Although the path of the query itself is controlled by the responses (positive responses cause the query to be forwarded; negative response do not), all reports are received by a central Detection Coordinator. This device correlates the information to synthesize a wider view that may already allow attack traceback through multiple administrative domains, if not from host to host. New traceback tools could be incorporated into such an architecture to allow for more robust traceback and finer resolution.

Stuart Staniford-Chen proposes a second possible architecture. In this architecture, all requests for traceback go to a central coordinator. The central coordinator queries the next possible link in the attack chain, processes the response, and issues new queries as needed. The central coordinator role is established within law enforcement agencies, and may comprise multiple levels of authority, for example, local, state, and federal. The queries are passed up and down the law enforcement hierarchy as the jurisdiction of the traceback developments. This is envisioned as a semi-automated interaction, with perhaps automated incident reports triggering the initial query and some responses generated automatically from system logs, etc. This semi-automated approach allows for an initial low level of automated capability growing into a more and more automated system as tools evolve.

Solution Constraints

Unlike passive defense or detection, attack traceback enters a realm where legal considerations must be taken into account. Even when liability or prosecution are not traceback goals, laws to protect privacy may limit the technical solutions. The three federal laws that dictate the legal considerations for traceback are the Electronic Communications Privacy Act (ECPA (18USC2701)), the Wiretap Act (18USC2511), and the Trap and Trace Act (18USC3121). Unfortunately, none of these statutes were written specifically with computer networks in mind, and the meaning of their provisions in this venue must be interpreted and tested in court. To date, insufficient case law exists to provide firm guidance.

The type of data used in traceback and the means used to collect it all have legal implications. For example, information gleaned from packet headers alone is fair game for traceback; legally, there is no expectation of privacy for packet headers. In contrast, packet contents are legally protected, and a traceback solution that uses packet contents may require lengthy and difficult legal procedures to obtain permission each time it is used. Gray areas exist for which the statutes themselves provide no guidance. For example, the content thumbprinting technique described in the section on Stream Matching uses digests of contents. Technically, the privacy of the original packet contents is protected, however, the legal status of this technique is currently undecided. In between packet header and contents in legal status is subscriber information (e.g., the name and address of an ISP subscriber). This type of information is legally protected, but can be obtained by a more simple procedure than content information.

Other legal distinctions are made among data that are merely collected versus data that are disclosed to others, disclosure of data voluntarily versus data whose disclosure is compelled by law, and access to stored data versus collection of data in real time. The meaning and impact of these distinctions will vary depending on whether the data are used in a civil action or a criminal prosecution. It is well beyond the scope of a technical paper to explore these ramifications thoroughly, however, two examples will serve to illustrate the complexity and ambiguity of the legal situation. In a state of affairs worthy of Catch-22, evidence gathered in anticipation of litigation may not be admissible as evidence. Data collected "in the normal course of business" is admissible. The exact boundaries of "normal course of business" are a gray area currently under-explored in case law. For example, is traceback information admissible if its routine collection is mandated by policy? What if the policy is driven by anticipation of litigating cases of intrusion? What if the data collection is not continuous, but automatically triggered by intrusive events? What if the trigger is not automatic, but via administrator

ISBN 0-7803-9814-9/$10.00 © 2001 IEEE

intervention? These and other related questions are currently open to interpretation.

Another legal gray area is encountered by querying traceback solutions. The ECPA makes it illegal for any government agency (not just those involved in law enforcement) to obtain electronic information from non-government entities without legal process (warrants, etc.). Suppose an IDIP query was initiated or passed along by a computer system serving a government agency? It is possible to interpret the law as forbidding this altogether. Another interpretation is that the query must identify its source as government/non-government, so that receiving devices can decide whether or not to respond. Still another interpretation is that as long as the query is confined to "have you seen this traffic", it is legal, but if it asks, "where does this traffic come from", it is not. The loosest (and most sensible) interpretation is that as long as the query arises from the system administration (automated or human) in the normal course of administering the government network, the ECPA does not apply. Eventually case law will sort out these options and establish some ground rules, but at this time the solution is not known. Another legal difficulty with querying systems is that eventually, in court, some human must testify as to the continuity of the query for each administrative domain it passes through. This requirement alone makes prosecuting cases based on a query-type traceback extremely expensive and difficult.

Some aspects of the technical solutions also have implications for the legal uses of traceback. Traceback solutions must incorporate features that allow time synchronization of events recorded at distant locations. Secure logging is necessary to protect evidence from attack in court. A legal case could be more easily built if all pertinent traceback information is captured in a single place, rather than having to be assembled from multiple logs and data files. Finally, since legal machinery moves slowly, the retention time of records is extremely important.

Perhaps even more daunting than the legal implications are the societal barriers to traceback. There is a surprising amount of suspicion that drives interactions between government and commercial entities, and between commercial enterprises. This lack of trust makes privacy of information a higher priority than attack traceback for many enterprises. As described in the Problem Space section, traceback will require additional infrastructure and cooperation among entities sharing the network. An important question for traceback is what business models are likely to cause the compliance needed to perform traceback.

There are two potential drivers towards increasing cooperation for traceback. The first is increasing government regulation that may force cooperation to some extent. The second is the increasing cost of attacks that may provide an economic motive for increased cooperation. Part of the cost of attack will be liability for the results of attacks. Companies may be sued for loss of privacy if private customer information is lost due to penetration. E-businesses may sue ISP's to recover the cost of lost business during denial of service attacks. Legally, businesses may escape responsibility for harm resulting from occurrences that could not be anticipated. Normally, criminal acts fall in this "unanticipatable" category, however there have already been legal rulings that network attacks are so common that they should be anticipated.

Eventually, the direct cost of attacks and the threat of liability for attacks may create demand for an "attack insurance" industry. Once insurance companies get involved, they will have a cross-enterprise incentive for attack traceback to allow for cost recovery. Premium incentives and conditions of insurance may be used to dictate adoption of standard attack traceback tools and techniques. To provide insurance against attack profitably, however, the insurance companies must have actuarial data. Thus, like so many other information assurance problems, the ultimate solution to attack traceback may rest on the definition of appropriate metrics and collection of data over a broad cross section of society.

REFERENCES

[1] S. Staniford-Chen and L.T. Heberlein, “Holding Intruders Accountable on the Internet”, *Proc. of the 1995 IEEE Symposium on Security and Privacy*, May 1995, Oakland, CA, pp. 39-49

[2] Stefan Savage and David Wetherall and Anna R. Karlin and Tom Anderson, “Practical network support for {IP} traceback”, *SIGCOMM*, 2000, pp. 295-306

[3] D. Song and A. Perrig, “Advanced and Authenticated Marking Schemes for IP Traceback”, *Proceedings of the IEEE INFOCOM01*, April 2001, Anchorage, Alaska

[4] W. Lee and K. Park, “On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack”, *Proceedings of the IEEE INFOCOM01*, April 2001, Anchorage, Alaska

[5] W. R. Stevens, *TCP/IP Illustrated Volume 1*, Addison-Wesley Publishing Company, 1994

[6] Y. Zhang, and V. Paxson, “Detecting Stepping Stones”, *Proc. 9th USENIX Security Symposium*, August 2000, Denver, CO

[7] S. Dietrich, N. Long, and D. Dittrich, “Analyzing Ditributed Denial of Service Attack Tools: The Shaft Case”, *Proceedings of the LISA 2000 System Administration Conference*, December 2000, New Orleans, LA

[8] S. Bellovin, “ICMP Traceback Messages”, *IETF Internet draft*, March 2000

[9] Jeff Rowe, “Intrusion Detection and Isolation Protocol: Automated Response to Attacks”, Presentation at RAID'99, September 1999