CERIAS Tech Report 2000-22

Packet Tracker Report 1

Thomas E. Daniels, Benjamin Kuperman,Clay Shields
Center for Education and Research in
Information Assurance and Security
&
Department of Computer Science, Purdue University
West Lafayette, IN 47907

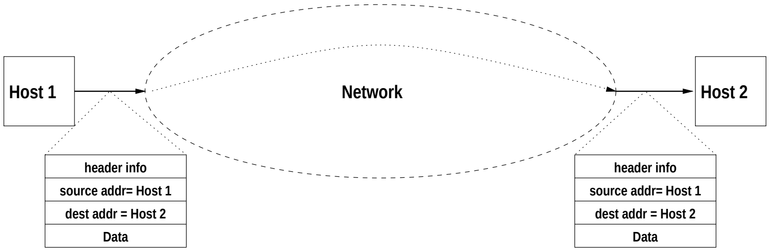

Figure 1: Simple Network Model

1 Introduction

When creating the suite of protocols that are used in the Internet today, the designers were more concerned with ensuring reliability and survivability than they were with providing accounting or security services [7]. This lead to a very simple network model. First, a packet-switched network was chosen to allow robustness and ease of routing around failures in the network. Second, all the network would provide was a simple packet-delivery service. This model was the basis for the Internet Protocol (IP), the fundamental protocol used in the Internet [16]. While in IP there are a few options for specifying a particular type of service requested from the network, and options to record the route the packet traveled or to mandate a particular route for the packet, all other services — including reliable transmission, congestion control and authentication of the source of a transmitted packet — have to take place at the endpoints of the communication [16]. Under this simple model, a host connected to the network gives a packet to the network, and the network attempts to deliver it to the given destination address. This is shown in Figure 1, in which Host 1 sends a packet to Host 2. As shown, the contents of the packet are some header information (including the packet and header lengths and checksum, the protocol being used and the type of service desired), the source of the packet and the destination for which it is intended, and data (which includes information not only for the application but also as necessary for multiplexing and reliability).

1.1 The Address Spoofing Problem

While this simple model formed the basis for the wide variety of successful services extant today, it is not without its flaws. Based on the information readily available, a host cannot be sure that a received packet has the correct source address. While in some cases the source correctness may be inferred from other data in the packet, particularly if some sort of strong authentication is used, it is typically very easy for some malicious sender to spoof the address of a packet that it sends. Figure 2 shows a malicious host sending a packet to Host 2 while pretending to be Host 1.

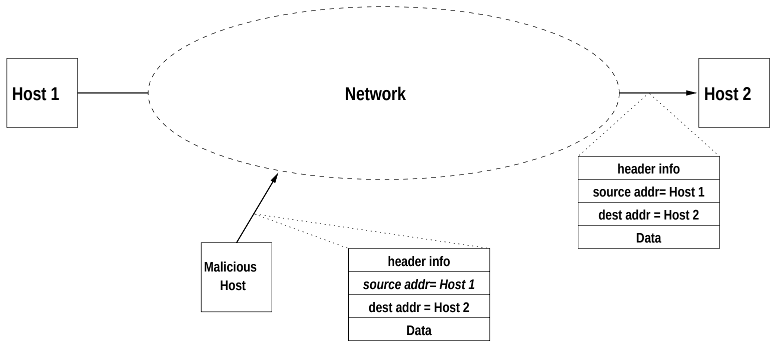

Figure 2: Address Spoofing

This capability has been used in a number of attacks, either to gain access to a host by exploiting a trust relationship it has with another host based solely on IP addresses [11, 1, 8], or to perpetuate a denial-of-service attack [4, 3, 5, 2]. In response, some individual domains have voluntarily added filters to their outgoing router that drop outgoing packets with external addresses. This prevents users inside the domain from spoofing packets by limiting the range of addresses that can be forged to those within that domain but does not prevent the use of address spoofing within the domain to hide an insider attack or to exploit internal trust relations.

It is also possible to prevent some of these attacks at the receiving end by requiring use of strong authentication, but that is not yet consistently feasible in practice, as it may be difficult to require such authentication for small packets as TCP SYNs. It can also be computationally expensive in terms of key management and key exchange. More importantly, while authentication will cause rejection of spoofed packets, it does not allow for discovery of the attacker, who is difficult to track and locate as the packet source address does not reflect any information as to his location. Finally, an attacker who has compromised a host may have access to the key information needed to defeat the authentication mechanism thereby leaving nothing more with which to trace the attack than in the unauthenticated case.

Currently, the main method used to locate such an attacker is to attempt to trace back the stream of forged packets while the attack is active [10, 17, 6, 13]. By following the stream of packets from router to router within the network it is possible to trace back and locate the particular source that might be conducting an attack. This is shown in Figure 3, where the internals of the network are revealed to be a number of *routers*, which are specialized hardware devices that do packet routing and forwarding, and the traceback occurs through the routers in the order they are numbered. This method is very limited, however, as it is necessary to have access to all routers along the path from victim to attacker, and this is often not the case. The attacker's packets may be traversing a number of domains under

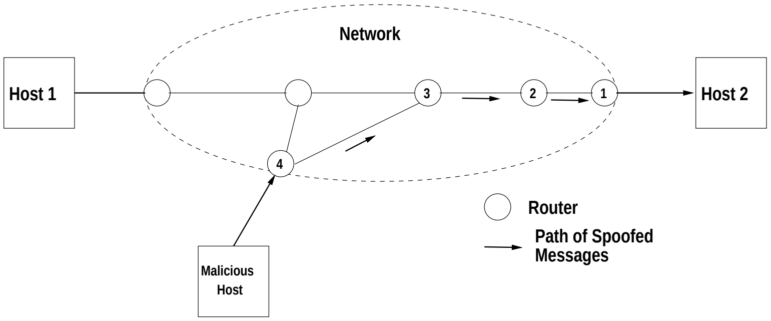

Figure 3: The Route of Traceback

different administrative control, in which case it is necessary to contact other network administrators, who have other demands on their time and may not be able to respond to an attack against a target for which they are not responsible. Additionally, this method is limited to tracing active attacks and thus must be done while the attack is occurring, or in the case of Intrusion Detection and Isolation Protocol (IDIP) [13], shortly thereafter. In all of these systems except for IDIP, no state is maintained in the network and therefore it is impossible to trace an attack after it has completed. In IDIP, a small amount of state is kept at special routers installed throughout the network that allows tracing of a packet immediately after its reception. It is unclear what level of state may be maintained without overburdening network components and how long this window of traceability is.

1.2 Traceback of Streams

An attacker may also take other actions to hide his location. A common (and unfortunately, often easy) way to do this is to compromise some remote host and use it to launch attacks. This makes it difficult to locate a particular attacker, because even if tracing back a stream of spoofed messages is successful, it results only in the location of the compromised host. The trace back then might have to be repeated if the audit data in the compromised host has been corrupted, or is insufficient to determine where the attacker came from. An attacker might use a series of compromised hosts, making the process of locating him very difficult, because hosts in multiple domains may be involved in different political regions around the world, or because the attacker may not be actively connected to the compromised host that is launching the attack, having set up the attack program to run after he has disconnected. Figure 4 illustrates how an attacker can use this method to hide their location. Notice that the data stream from the attacker passes in and out of the network at several different places.

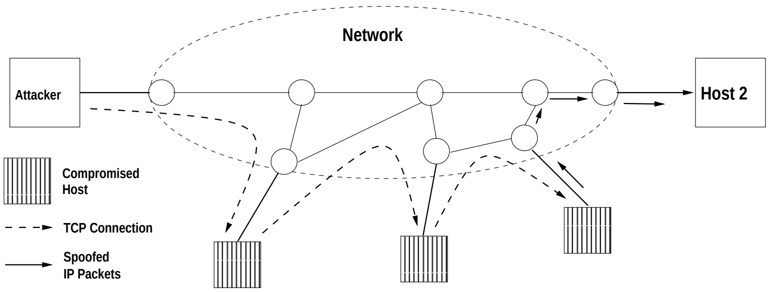

Figure 4: Using Compromised Hosts to Conceal Origin

1.3 The Problem

In each of these cases an attacker is aided by the fact that the network and its hosts do not deliver or maintain any information about the traffic carried. Packets are not delivered with any information about their originator, and this results in an attacker being able assume the identity of others, in terms of sending packets, with relative impunity. Data streams are traceable only while they are active, and once ended are impossible to follow, making it possible for an attacker to escape without his location being detected.

1.4 Past Work

There is previous and current work that attempts to characterize how a network device that sits at the edge of a particular network (either at the sub-net the host lies on, or at the boundary between autonomous domains) can attempt to match an incoming and outgoing stream so as to detect when an attacker is using a compromised host for forwarding. This serves two purposes. First, it allows for detection of compromised hosts. Second, it allows for a "shortcut" in any attempt to trace back a stream to an attacker. While most of these previous efforts have been attempts to detect an attackers activity in real time [10, 17, 6], some work has been done on recording and providing a fingerprint of a data stream, so that streams monitored in different locations around the network can be compared after the fact [14, 15].

Other work presents a host-based approach to tracing an attacker who is logged on through a number of hosts[9]. During the process of logging into a remote host, the originating host presents a trace for the user showing the hosts he has traversed and user names used on those hosts. The destination host then takes steps to verify that the user is actually logged into those hosts. If the verification step succeeds, the login is allowed. In either case, the trace is logged for later use by an administrator. In tightly controlled environments this may prove to be a useful approach, but it may be subverted using covert channels and other tricks.

Previously researched solutions are mostly unproven in real networks and have many problems that limit their utility. For instance, it is unclear how commonly false matches will occur in the fingerprinting techniques.[15, 14, 10]. Additionally, these techniques are susceptible to link-based encryption and evasion techniques similar to those described by others. [12] None of the techniques have addressed the problem of interdomain tracing nor incorporated measures to help assure the privacy of users. Also, none of the prior work has looked at limiting the ability to trace connections to authorized individuals. Finally, most of these traceback techniques only work for active attacks, but often attacks are not detected until it is too late to launch a trace during the attack.

Tracing packets and streams in a variety of network environments is an important component of the fledgling field of network forensics. The techniques for traceback systems proposed so far are only applicable to closed, tightly controlled environments. For traceback systems in open networks like the Internet, we must address the problems of privacy, trace integrity, and passive tracing. Furthermore, we must evaluate existing fingerprinting techniques for use in large, highly-connected networks and develop better techniques if necessary.

2 Environments

When examining possible systems of network traceback, it is important for us to define the environment in which our solutions will be applied. We suggest that there are two defining characteristics for solution environments: who controls the hosts and who controls the network.

Centralized host control implies that a single administrative authority is able to define and control all of the participating hosts on a network. This authority can determine the hardware, operating system, software installed, network services offered, and has the ability to customize or modify the network applications in any way they desire, provided the network will still carry their data.

Diverse host control implies that there is no central authority that can control and regulate the hosts connected to the network. No guarantees can be made about the specific hardware, software, operating system, or network services that hosts offer. Subsets of hosts might be under a single control, but not necessarily the entire set.

Local network control implies that all of the network infrastructure is under a single administrative domain. This administration can dictate the hardware, topology, and routing used in the network. This administration also has the authority to change network protocols,

	Network Control	
	Local	**Diverse**
Centralized Host control	Closed Model	Intranet Model
Diverse Host control	Academic	Internet

Table 1: A matrix of various computing environments encountered in a networked system of computers.

modify or examine any data flowing over the network, and regulate who or what is connected to the network.

Diverse network control implies that no central authority exists that controls the network infrastructure. Standard network protocols are required, or else data may or may not be routed. It is not possible to make any guarantees about who sees any data, or who the sender of any data is.

As seen in table 1, these four characteristics serve as the primary distinguisher for our four network models. The labels we have selected are arbitrary, and intended to convey the general concept of each environment.

Closed Model - This is an environment where the network hardware and all machines connected to it are under a single administrative control. This environment allows arbitrary changes to be made to network protocols and end machines. All of the packets viewed on the network should have been generated by a machine under the administrative control, and the packets never cross "untrusted" hardware.

Intranet Model - This environment is a collection of LANs that are interconnected by some form of "secure" connections (e.g. VPN tunnels, leased lines, etc.). Packets travel between clusters of machines across a shared network. A single entity can be controlling these clusters of machines, but it is necessary that the data be able to be carried over a public network if necessary, so the freedom to modify network topology, hardware, or protocols is lacking.

Academic Model - This environment is the situation on many University campuses. A single network connects the various machines on the campus, however, the machines are not centrally administrated. Any changes in the network protocols requires consensus building amongst the diverse groups. It is assumed that machines can be connected to the network at any time, and that they may or may not be well behaved.

Internet Model - There is a collection of LANs, WANs, and single hosts all sharing a network structure that does not have any central controlling authority.

The primary factors described above are not the only factors that define the environment in which traceback solutions are implemented. The following factors describe some of the various issues that also need to be considered, but are not fundamental to the enviroments in our discussions.

Resources of an entity are composed of three distinct subcomponents:

1. **Financial resources** describe the ability of an organization to purchase or otherwise expend money. This resource is used to augment and offset any deficiency that might exist in the other types of resources. Depending on the organization, the use of financial resources might be tightly controlled and/or under specific restrictions. For instance, the DoD can spend its allocated resources with relative freedom, but a small business might have to justify every single dollar of expenditure and use of this resource would be difficult.
2. **Technical resources** are sources of technical knowledge and expertise. If an organization does not naturally have a large technical reserve to draw on, they can expend financial resources to improve it by either hiring new staff or consultants. A University might have restrictions on the expenditure of financial resources, but they have a vast technical resource in their professors and graduate students.
3. **Manpower resources** expresses the ability for work to be done. A small company has limited manpower resources, while a university has a large set of manpower (students!). Again, any lack in this area can be offset by the expenditure of financial resources.
4. **Infrastructure resources** are the various computer and networking hardware available to an organization. This can be expressed in terms of bandwith, CPU power available, and hardware availability. An infrastructure rich environment such as the labs at Sun Microsystems or Cisco can build custom hardware to meet task needs, while an infrastructure poor organization, like a public school, would be hard pressed to meet current user needs.

Expectations of Privacy are defined in respect to some outside party. Users in an network environment might have no expectation of privacy as in a corporate setting where the users sign away their privacy. Another example is a classified computing environment where a user expects privacy from his peers so as to maintain confidentiality but certainly not from management for reasons of oversight. Users may also expect privacy from host and/or network administrators in some environments.

Societal Cost is a term that we use to describe the various incentives that exist for needing a network connection to be traced. In an intelligence agency, being able to trace an attacker's connection might be an issue of life and death of operatives. In a university setting, the issue might be to track an attacker that is using university resources to launch attacks. A provider of high bandwidth communication channels might not have any desire to trace connections, but simply need to bill the proper customers based on traffic.

The model environments described above serve as a baseline for evaluating and designing traceback systems. The defining characteristics of the environments limit the possible solutions for a given environment, and the secondary factors help us to further describe the approaches based on their social, financial, and practical impacts.

3 Conclusions

There are two main problems that make tracing network traffic to its source difficult: address spoofing and the redirection of traffic through multiple possibly compromised hosts. Each of the existing traceback techniques addresses a small part of the overall problem space, but they fail to address many issues needed in a useful traceback system such as privacy, In order to evaluate possible solutions we will consider solutions in terms of appropriateness for a model environment and also using various secondary factors. Finally, to achieve our goal of improving the state of the art in traceback systems, we must address the issues left unsolved by existing techniques and develop solutions with them in mind that are compatible with the various model environments.

References

[1] Bellovin, S. M. Security Problems in the TCP-IP Protocol Suite. *Computer Communications Review 19*, 2 (April 1989), 32–48.

[2] CA-96.21, C. A. TCP SYN Flooding and IP Spoofing Attacks. http://www.cert.org/advisories/CA-96.21.tcp_syn_flooding.html, September 1996.

[3] CA-97.28, C. A. IP Denial-of-Service Attacks. http://www.cert.org/advisories/CA-97.28.Teardrop_Land.html, December 1997.

[4] CA-98.01, C. A. 'Smurf' IP Denial-of-Service Attacks. http://www.cert.org/advisories/CA-98.01.smurf.html, January 1998.

[5] CA-98.13, C. A. Vulnerability in Certain TCP/IP Implementations. http://www.cert.org/advisories/CA-98-13-tcp-denial-of-service.html, December 1998.

[6] Chang, H., and D.Drew. DoSTracker. This was a publically available PERL script that attmepted to trace a denial-of-service attack through a series of Cisco routers. It was released into the public domain, but later withdrawn. Copies are still available on some websites., June 1997.

[7] Clark, D. The Design Philosphy of the DARPA Internet Protocols. In *Proc. ACM SIGCOMM* (August 1988), pp. 106–114.

[8] Joncheray, L. Simple Active Attack Against TCP. In *Proceedings of the Fifth USENIX UNIX Security Symposium* (Salt Lake City, Utah, June 1995).

[9] Jung, H. T., Kim, H. L., Seo, Y. M., Choe, G., Min, S. L., Kim, C. S., and Koh, K. Caller id system in the internet environment. In *UNIX Security Symposium IV Proceedings* (1993), pp. 69–78.

[10] Mansfield, G., Ohta, K., Takei, Y., Kato, N., and Nemoto, Y. Towards Trapping Wily Intruders in the Large. In *Proceedings of the Second Annual Workshop in Recent Advances in Intrusion Detection(RAID)* (West Lafayette, IN, September 1999).

[11] Morris, R. A Weakness in the 4.2BSD Unix TCP-IP Software. Tech. Rep. 17, AT&T Bell Laboratories, 1985. Computing Science Technical Report.

[12] Ptacek, T. H., and Newsham, T. N. Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection. Tech. rep., Secure Networks, Inc., Suite 330, 1201 5th Street S.W, Calgary, Alberta, Canada, T2R-0Y6, Jan. 1998.

[13] Rowe, J. Intrusion detection and isolation protocol: Automated response to attacks. Presentation at RAID'99, Sep 1999.

[14] Staniford-Chen, S., and Heberlein, L. Holding Intruders Accountable on the Internet. In *Proc. of the 1995 IEEE Symposium on Security and Privacy* (Oakland, CA, May 1995), pp. 39–49.

[15] Staniford-Chen, S. G. Distributed tracing of intruders. Master's thesis, University of California, Davis, 1995.

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

[17] Zhang, Y., and Paxson, V. Stepping Stone Detection. Presentation at SIGCOMM'99, New Areas of Research, August 1999.