Network Traffic Tracking Systems: Folly in the Large?

Thomas E. Daniels[*]
Center for Education and Research in
Information Assurance and Security (CERIAS)
1315 Recitation Building
Lafayette, IN 47907-1315
daniels@cerias.purdue.edu

Eugene H. Spafford
Center for Education and Research in
Information Assurance and Security (CERIAS)
1315 Recitation Building
Lafayette, IN 47907-1315
spaf@cerias.purdue.edu

ABSTRACT

Recent distributed denial of service attacks have demonstrated the difficulty with tracing network attackers on the Internet and simultaneously led to calls for development of systems to track network traffic to its source. Tracking network traffic is difficult because of two basic techniques used to obfuscate the source of the traffic: spoofing and redirection. In this paper, we examine the desirable properties of network traffic tracking systems (NTTS) from both the technical and social perspectives. An analysis of the feasibility of a system with these properties in a number of increasingly open network models leads us to a number of conclusions. First, NTTS may be very successful in relatively closed environments where there is strong control of the infrastructure, and there is no expectation of privacy. Second, in an open, global Internet, it is not be feasible to deploy a perfect NTTS. Third, if a perfect NTTS for the Internet is not possible, how do we evaluate the consequences of deployment of an evadeable NTTS.

[*]Primary and corresponding author.

1. INTRODUCTION

Recent distributed denial of service attacks have demonstrated the difficulty with tracing network attackers on the Internet and simultaneously led to calls for development of systems to track network traffic to its source. We call these proposed systems, network traffic tracking systems (NTTS).

We consider the area of network traffic tracking a new paradigm in network security because there has been little past work, to some extent, we are going against the relatively well researched topic of network anonymity, and we feel the problem is not well characterized. The majority of the past work[7, 15, 19] is directed at tracing spoofed packets and denial of service attacks. Only three known works[13, 20, 21, 23] address the tracing of attacker's actions through the network. Another factor is that while there are a substantial number of works that aim to anonymize network access, no work discusses ways to do the inverse. Finally, instead of dwelling on one or two well known problems, we hope to consider the larger problem of network anonymity and what are consequences of systems that attempt to eliminate it.

The development of NTTS is not just a technical issue. There are issues of privacy and control of the system. In a system that covers the Internet, how does its multinational nature and distributed control make an effective NTTS possible? Furthermore, if an NTTS is less than perfect, how do we justify the cost of a system that only catches the dumb criminals?

We feel that this is an excellent topic for the new security paradigms workshop for a number of reasons. First, it is a very young area to which few in the field have given much thought. Second, there is need for consensus building around terms and the desirable properties that make up the area. Third, our submission is likely to create a good deal of debate about the need for systems which are likely to be fallible for the foreseeable future. Finally, there is some merit to the idea that we do not need more secure networks, rather we need methods of determining who is accountable for an action on the network thereby deterring attacks in the first place.

2. ANONYMITY IN THE NETWORK

Informally, anonymity in the network is achieved by using two basic methods: spoofing and redirection. Spoofing in this context means to lie about the source of some piece of network traffic. Redirection means that a network entity receives network traffic, possibly modifies it in some way, and then resends the traffic. These methods may be used at various levels of a protocol stack in order to obfuscate the source of the traffic that the protocol carries.

Spoofing attacks are most closely tied to attempts to exploit trust relationships where anonymity is simply a by-product of the attack [16, 1]. Spoofing is used for anonymity in many denial of service attacks including the recent distributed denial of service attacks[11, 9] and classic ones[4, 6, 5, 3] as well.

Redirection is also used to hide the source of network traffic. Legitimate systems such as Crowds[18] and Onion Routing[17] use it along with encrypting the contents the traffic to decouple the true source of the traffic from the receiver and possible eavesdroppers. Network attackers use a common form of redirection to hide their network access point. To do this, the attacker logs into a number of hosts in a serial fashion so that their user session is redirected from one host to the next. The attacker then launches his attack from the final host in the chain so that the source address of the traffic is that of the final host. To make it even more difficult to trace, the attacker may use subverted hosts in many different jurisdictions and may delete useful logs from the hosts.

2.1 A Simple Model of the Problem

We have developed a high-level model of anonymity in the network that illustrates the depth of the problem and unifies the many instances of the problem observed by others. It should be known that this model is a work in progress and that we are continuing to further formalize it.

We introduce the model by recalling the OSI Protocol Reference Model. For a good treatment of the model, consult Tanenbaum's textbook on networking[22]. A protocol stack is made of a number of layers each using the services of the layer below to provide services to the layer above. In the OSI model, the protocol stack is composed of 7 layers from the physical layer up to the application layer. In the stack, entities at layer n on one node use the services at layer $n-1$ and below to transfer flows of information to other layer n entities on some other node.

As seen in Figure 1,We base our model on the OSI Reference Model, but we add a layer to the top called the User Session Layer. The User Session Layer models the behavior of user login session in which a user logs into a node by way of some application, performs some action, and eventually logs off. As will be described later, this addition allows us to model so-called "island hopping" where an attacker logs into a number of hosts in serial in order to hide his identity. In the case where a flow neither originates with nor is bound for the current node, the flow may be forwarded (and possibly modified) by some level of the protocol stack towards its destination. We call the mechanisms that handle redirections, relays. Redirection of traffic seems to be the fundamental feature of computer networks and also appears to be the fundamental method for gaining network anonymity. The relay can make a flow's source difficult to determine by replacing the source information with its own as it relays the traffic. Determining the source of a flow is also made difficult because a relay may provide no backward looking information When looked at from this perspective, spoofing is not really the problem since anyone trying to gain anonymity will not volunteer their identity if possible.

Somewhat more formally, we define a relay to be an entity at some level of a protocol reference model that accepts a

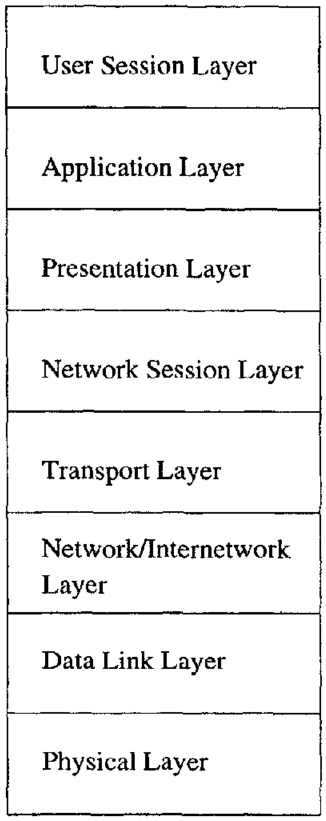

Figure 1: Our modified protocol reference model including the User Session Layer.

flow from the network, possibly modifies it, and passes it on to another node on the network. At any given time, a relay may directly accept traffic from some *input set* of nodes and output to some *output set* of nodes.

We will now demonstrate how the relay concept applies to several known anonymization techniques used in networks. An attacker uses IP spoofing to anonymize IP packets by simply setting an incorrect source IP address in the header of the packet and then sends the packet across the network. In this case, the routers that transfer the packet are modeled as relays. The relays just forward the traffic on to their neighbors according to their routing tables. As the packet is forwarded through more routers, the set of possible ultimate origin nodes becomes larger. To see this, consider a packet outward bound at an organization's border router. Most likely, the packet must have been sent by node within that organization. However, The same packet observed two hops further inside the organization's Internet service provider (ISP) may have been sent from any host serviced by the ISP.

So called island hopping attackers hide their source by logging in to a number of machines in serial fashion. The relays here occur at the User Session Layer described above. Anonymity is gained because the user appears to be coming from the last host in the chain, and it is likely that the input set for relay is quite large. Another similar approach used by attackers is to an application specific proxy. In this case, one or more Application Level relays are established and used in series. Finally, anonymity systems such as onion routing use a combination of Application Level relays and Session Layer relays that cryptographically modify the flow as it is forwarded.

To see just how difficult a problem an NTTS can face, consider that many techniques can be used together at different layers of the protocol stack. Clearly, new research needs to be done to more fully understand these problems and how to better address them. We plan to continue to formalize this model with the hopes that it will guide future development of NTTS and allow us to make broad claims about network anonymity and systems that deal with it.

3. PRIVACY IN THE REFERENCE MODEL

One observation that we have made is that as the protocol stack level of the relay increases towards the top, the information involved becomes more privacy sensitive. To see this, consider an NTTS that is operating at the Network Layer to trace spoofed packets. Such a system need not be concerned with anything but enough information to identify a packet and the address from which the packet originated. In fact, the contents of the packet could be encrypted and it should not matter to the NTTS. In contrast, an NTTS built to trace through application or anonymization proxies is in an ideal position to detailed personal information such as profiles of web sites visited, newsgroup subscriptions, and electronic mail contacts.

4. DESIRABLE PROPERTIES OF AN NTTS

This section tries to list the desirable properties of an NTTS. For each there is a short discussion of its importance. Many of these properties are clearly not orthogonal to each other in the sense that there are tradeoffs between them. For instance, a system that is less precise may be more accurate because it makes fewer mistakes.

4.1 Accuracy

Accuracy is certainly a desirable property of an NTTS. In this case, we define the accuracy of an NTTS as the probability that the source found for a certain piece of network traffic will be correct. The accuracy of an NTTS may have deep consequences for its use. For instance, a highly accurate system might be sufficient to get a search warrant when used to trace a widespread denial of service attack, but less accurate system may not be sufficient. The issue is how accurate must such a system be and how do we determine its accuracy in the real world.

4.2 Precision

Even if the NTTS is highly accurate, it does not mean the NTTS always finds the source with the same level of specificity. We call precision the level of specificity with which an NTTS can determine the source. For instance, the source could be specified as a process, host, or subnetwork. Alternatively, it could even be in terms of physical location of the source[10]. Precision might vary widely even within the same system and between traces depending on the steps taken by an individual to hide his tracks. For instance, a system might be thwarted by a savvy user using onion routing[17] system and therefore only be able to link the trace to any user of said system. Alternatively, the same system might easily trace a user to the host he was using.

4.3 Resist Subversion

To be accurate and precise, an NTTS must resist subversion by those wishing to hide the source of their traffic (and others). After all, if someone does not wish to hide the source of traffic, they are free to use the network in the usual manner. The notion of an NTTS is not meaningful unless there are methods to lie about or misrepresent the source of network traffic.

4.4 Low Overhead

An NTTS used in the real world might consume considerable network and other resources. These resources include network bandwidth and processing time and storage (volatile and non-volatile) on network components.

It is desirable that an NTTS consume as few of these resources as possible. Consumption of too much bandwidth by an NTTS will limit the usefulness of the network. If an NTTS uses excessive processing power on a router, it may cause dropped packets or slow the switching process thereby adding latency to the network. Interestingly enough, adding latency is the reason often given for not enabling egress filtering to prevent the spoofing of packets. An NTTS that requires large amounts of volatile storage may add cost as discussed below or cause network components to operate more slowly while coping with the increased demand. An NTTS would likely use non-volatile storage for keeping long term records of the origin of traffic. It is important that an NTTS store the origin information succinctly because the amount of non-volatile storage available along with the amount of traceable traffic dictates the traceability of older traffic.

Another important aspect of resource usage is how the resource is used. For instance, an NTTS may consume some resource for every TCP connection sent on the network. We call this *traffic-based usage* because the resource usage is a non-constant function of the traffic on the network. Alternatively, a resource may be used only when doing a specific trace and not for every piece of traffic being passed over the network. We call this *trace-based usage.* Most conceivable and known NTTS will use resources in both of these manners but may make different trade-offs between the two.

4.5 Low Cost

The cost of implementing and operating an NTTS should be as low as possible. If a solution is too costly, then it may never be implemented or only implemented as costs for components drop.

4.6 Scalability

Scalability is important in any widely deployed NTTS. Proposed solutions work to some extent, but would be too expensive or infeasible in very large networks. Along the same lines as scalability, many suggested NTTS fail to consider partial deployment. If a system requires that all existing infrastructure be modified before becoming useful, it is unlikely to be deployed. For this reason, an NTTS should work without total deployment although likely with some caveats such as reduced precision or accuracy.

4.7 Realtime

A realtime NTTS requires that the network traffic to be traced is being received during the trace. In order to use a realtime NTTS, one must first detect the traffic, decide that it is worth tracing, and do the trace. It is desirable that an NTTS support non-realtime tracing so that an administrator may determine the source of network traffic for some time after it is received. This allows for after the fact tracing of an attack that was not detected while it was in progress. One obvious problem with supporting non-realtime tracing is that it requires storage of trace data which consumes storage and possibly bandwidth resources.

4.8 Privacy and Control

The social requirements of a widespread NTTS are primarily privacy and control. Privacy matters revolve around what information is stored about individuals, their network traffic, and consequently can be inferred from the traffic. Another issue of privacy is not only what data is stored but where it is stored and for how long. There is the obvious tradeoff of the non-realtime NTTS need for storing data for long periods of time versus the increased risks to privacy that such data storage poses.

Control of an NTTS refers to both a mechanism for controlling access to those authorized to trace network traffic and an authority or authorities consisting of people who manage this mechanism and determine who is authorized. There are several possibilities here including government, organizational, and recipient control. Governments and organizations may control NTTS's, but what does this mean when the traffic extends beyond the jurisdiction of these entities? If a government controls the system, what is needed to do a trace? Recipient control is an attractive approach considered in some recent work[2]. In this case, the recipient of the traced traffic flow receives the trace data and hence controls access to it. While this may be preferable for an NTTS geared towards tracing distributed denial of service attacks[2], it may compromise an email anonymizer for use by whistle blowers.

5. ACHIEVABILITY OF OUR DESIRABLES IN MODEL ENVIRONMENTS

In the following, we rudimentarily describe three basic model environments[8] in terms of the ability of a managing organization to control the hosts and the network that connects them. We also consider the achievability of our desirable features in each of these environments.

5.1 Closed Model

The closed model is managed by a central authority that controls both the hosts and the networks within it. It is not connected to networks outside its control. The authority may customize host software and operating systems, mandate network topology, and modify network components as it sees fit.

In this model, we are free to modify the hosts and networks to maintain and report the information needed to track network traffic at all levels. There is control over the origination point of the traffic, the network it travels over, and the recipient.

In this model, it is fairly clear that a precise and accurate NTTS may be developed to track network traffic. By controlling the behavior of hosts in this environment, it should be possible to mark network traffic with its origination point when it is sent and then log the information on receipt. If sufficient audit trails are available, it may even be possible to link traffic to a specific user identifier. This is one possible solution that could be implemented by simply requiring protocols such as IPsec with authentication[14].

If we are free to modify this host as necessary, overhead should be negligible and limited to the storage needed to maintain an audit trail. Similarly, cost of implementation may be high, but operational costs should be negligible. Scalability is not so important as closed networks are probably limited in size.

The issue of privacy in the closed network is probably not very important because users are unlikely to have any expectation of it. They are likely to be employees of the central authority and may have signed away such expectations away in advance. Control of the tracking information may nevertheless be sensitive, and we have a central authority that can manage access in this event.

Tracing high level flows in the closed model is probably the easiest of those we present. This is because we control the end points and can therefor modify even high level protocols or use approaches similar to the Caller Identification System for the Internet[13]. In this case, a recursive version of the ident[12] protocol is created for tracing user sessions that are anonymized by island hopping. Similarly, low level flows should pose no more (probably less) difficulty to an NTTS in a closed environment.

5.2 Academic Model

The academic model is one where there is a central authority that has control of the network that connects hosts or small subnets of hosts but not of the hosts themselves. One possibility that has been suggested for such situations is to modify network traffic as it enters the network to indicate its source.

In this model, we still control the network infrastructure so we may be able to trace with subnet-level or possibly host-level precision. Since the user may be able to forward traffic through hosts with arbitrary modifications (such as encryption), it may be very difficult to trace traffic to its ultimate source. Because of this, the accuracy of network-oriented systems will rely on the frequency of use of such obfuscation techniques and our ability to build systems that can correlate traffic that is being forwarded in such a manner.

Overhead of an NTTS in this scenario would likely be quite high in terms of processing time in routers and other network components. Such components would necessarily cost more than conventional routers. It is unclear how much bandwidth would be consumed in this situation.

Scalability becomes important in this model, but it is likely that most practical solutions would scale to such an environment, but possibly to one no bigger.

Privacy becomes an issue in the academic model. It is likely that there are multiple groups of users and that those users have differing expectations of privacy. We do have a central authority that can control access to private information.

In the Academic Model, tracing lower level flows, such as at the Network Layer, should remain possible since we control the internetwork. However, it may be considerably more difficult to trace higher level flows because we do not control the hosts which make useful relays for the higher level protocols. Work such as that by Staniford-Chen[20] try to overcome this but are easily evaded. Future work is important to determine how reliably and under what conditions we can trace high level streams using network control alone.

5.3 Internet Model

The internet model is one where no one authority controls the hosts or the network. There are many uncooperative authorities in control of relatively small sections of the network. The subnetworks all rely upon common, standardized protocols to interact with each other.

It is unclear that we can build a highly accurate and precise NTTS in the wide open internet model. The only thing that is shared among them is the set of shared internetwork protocols. We must therefore consider modifying the internetwork protocols to provide traceability.

Issues of cost and overhead are not clear cut in this case. If the protocols require replacement of network components such as routers then the cost would be quite substantial. Similarly, the new protocols should not significantly strain the routers as this would add overhead to the network in terms of latency and bandwidth.

The problem with modifying the internetwork protocols is that in order for the changes to be practical, the solution must scale to millions of networks and must be partially deployable. Such a protocol modification must be lightweight and store trace data at the end points in order to scale well and provide an after-the-fact solution. Also, it is impractical if not impossible to upgrade all components on the Internet at once and so the solution must be partially deployable.

The problem with a partially deployable NTTS based on modified protocols is that it may be difficult to compel all networks to use the new protocols. This will inevitably create safe havens running the older protocols through which users can redirect their traffic thereby defeating the NTTS and leading to accuracy problems. Even if a router or other network component implements an NTTS, there is no guarantee that the component will actually run the NTTS. If one considers the likelihood that an attacker is more likely to control network components nearer him, we can see that it will be very difficult to increase precision in this model.

The internet model makes the privacy issues more compelling. The answer to the questions of what data is stored and where will be dependent on the NTTS, but it is likely that the more workable solutions will deliver data about a path to the recipient of the traffic. This leaves the door open for abuse by sites that receive large volumes of user traffic. Instead of being used for tracking attackers, the data might be used for discrimination, marketing, or user profiling.

If the receiver of the traffic is not viable for storage and control of privacy related data, some other authority might be then considered. The problem is that in the internet model, who would we give that authority? Clearly, the multinational nature of the Internet makes any one government inappropriate for the task. Also, it is not uncertain that a central authority is feasible for such a task. It may be that a disinterested central authority can control access to the data without actually storing the data itself. For instance, the authority might coordinate and store cryptographic keys used to secure the data stored around the network but not store the data itself. More research needs to be devoted to maintaining privacy in the presence of an NTTS.

In the Internet Environment, the only guaranteed commonality among the nodes is their use of standardized network protocols such as IP. We are beginning to see modifications for IP that support an NTTS for specialized types of flows such as IP flooding attacks[2]. This is tenable because the work addresses a very specific type of flow and takes advantage of its volume. The technique is also tailored to minimize resource usage in routers and hosts. It is doubtful that higher level protocols can be handled in as lightweight a manner as this.

6. CONCLUSIONS

We have given a high level overview of network traffic tracking systems and the problem of network anonymity. By introducing a simple model of the problem, we have shown that the problem space is much more rich than anecdotal evidence would suggest. We have listed and discussed the desirable properties of such systems and analyzed them to some extent in three model network environments.

It appears that development of a useful NTTS is considerably easier in more closed environment where greater control of infrastructure can be had. More open models show us that the problems become more difficult and may require changes to basic protocols to implement traffic tracking.

Another interesting dimension of this discussion is how our analyses vary in each environment as we consider the difficulty of tracing higher level flows versus lower level ones. As the environments become more open, higher level flows seem to become more difficult.

It appears that in the arbitrary world of the Internet, NTTS's will remain subvertible and quite possibly an affront to privacy. We must therefore ask ourselves if the benefit of a subvertible NTTS in catching sloppy or ignorant attackers can be offset by the substantial costs and privacy risks inherent in such systems.

7. REFERENCES

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

[2] S. M. Bellovin. Icmp traceback messages: Internet draft. `http://www.ietf.org/internet-drafts/draft-bellovin-itrace-00.txt`, March 2000.

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

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

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

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

[7] H. Chang and D.Drew. DoSTracker. This was a publically available PERL script that attempted 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.

[8] T. E. Daniels, B. Kuperman, and C. Shields. Packet tracker technical report 1. Technical report, Center for Education and Research in Information Assurance and Security (CERIAS), Purdue University, 1999.

[9] D. Dittrich. The "tribe flood network" distributed denial of service attack tool. `http://packetstorm.securify.com/distributed/tfn.analysis.txt`, October 1999.

[10] J. Glasner. Feds ok cell phone tracking. *Wired News*, September 1999.

[11] C. I. N. IN-99-07. Distributed denial of service tools. `http://www.cert.org/incident_notes/IN-99-07.html`, November 1999.

[12] M. S. Johns. Rfc 1413: Identification protocol. `http://docs.unode.net/rfc/html/rfc1413.html`, February 1993.

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

[14] S. Kent and R. Atkinson. Ip authentication header. Request for Comments 2402, November 1998.

[15] G. Mansfield, K. Ohta, Y. Takei, N. Kato, and Y. Nemoto. 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.

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

[17] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Proxies for anonymous routing. In *12th Annual Computer Security Applications Conference*, pages 95–104. IEEE, December 1995.

[18] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. Technical Report 97–15, DIMACS, April 1997.

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

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

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

[22] A. S. Tanenbaum. *Computer Networks*. Prentice Hall, Inc., 2 edition, 1988.

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

ACKNOWLEDGEMENT

Because of illness, Clifford Kahn, the author of the following paper, could not be present at the Workshop. On the eve of the session, Mary Ellen Zurko volunteered to present her interpretation of the paper and to lead a discussion on its concepts. Bob Blakley, in his continuing rôle as NSPW Scribe, transmitted his notes of the resulting discussion to the author. The paper that follows reflects the discussion and the important 'collaborative' contribution made by Ms Zurko and the workshop participants.

-- Ed.