Network Forensics: Towards a classification of traceback mechanisms

Sarandis Mitropoulos, Dimitrios Patsos, Christos Douligeris
Department of Informatics,
University of Piraeus,
80, Karaoli and Dimitriou Street,
Piraeus,
GREECE
sarandis@unipi.gr, dpat@space.gr, cdoulig@unipi.gr

Abstract

The traceback problem is one of the hardest in Information Security and has always been the utmost solution to holding attackers accountable for their actions. This paper presents a brief overview of the traceback problem, while discussing the features of Software, Network and Computer Forensics. In the rest of this paper, various traceback mechanisms are examined while categorized according to their features and modes of operation. Finally, we propose a classification schema for all traceback methods in order to assess and combine their benefits so as to provide enough information for Digital Forensics analyses, thus getting -the right way- one step closer to the actual attacker.

1. Introduction

The Internet has evolved through the years after the convergence of Information Technology and communication systems as the current communications medium for the whole planet. In parallel, malicious users have also evolved from launching simple cracking attacks to extremely complicated Reflective Distributed Denial of Service attacks [1].

From the Information Security Management view, Incident Handling and Response has always been an important element in every Security Policy developed in corporate, academic or government environments [2]. From the legal point of view, Digital Forensics techniques and methodologies are dealing with the critical issue of mirroring computer commands issued by attackers to electronic crime actions and charges. From the security research point of view, a series of traceback mechanisms have been developed during the last few years, in order to find the source of the attack traffic in (near) real time, by automating the process of reconstructing the attack path.

Let C = h_1+h_2+..+ h_i + h_{i+1} + … h_n be the connection path between hosts h_i (i=1,..n). Then the traceback problem is: given the actual IP address of host h_n to identify the actual IP addresses of h_{n-1}, … h_1.

This paper first presents a very short overview of Digital Forensics techniques while reviewing the most well-established automated traceback mechanisms and exploring their characteristics. Then we introduce a classification, through the use of multiple dimensions of the technologies and the possible applications. Limitations or features of the presented traceback schemas are also presented. Our main goal contribution is through the combination of traceback mechanisms (to extend their limitations of reaching the attacker's network) with Incident Response and Digital Forensics actions in an automated way, in order to reach to the actual attacker. Open issues towards this path that require extensive reviewing and testing are analyzed.

2. The Problem Space

Many masquerade techniques can be used in order for an attacker to hide his/her original identity. Masquerade (aka impersonation) attacks are nothing new to Information Security. They can be reproduced in various ways by the attackers, mostly by using the following techniques:

- Link Layer spoofing, also known as MAC address spoofing (e.g. using a different MAC address than the original) [3].
- Internet Layer spoofing, also known as IP Source address spoofing (e.g. using a different source IP address than the original) [4].

**Corresponding author*

- Transportation layer spoofing, also known as port spoofing/port forwarding (e.g. using a different TCP/IP port than the original one).
- Application layer spoofing (e.g. using a different email address than the original).

In a typical DoS attack [5], a number of attackers use some intermediate hosts and networks (i.e. different routing paths) in order to launch an attack to a victim machine(s).

We present a conceptual attack path in figure 1, where a_i are the attacking hosts, R_i are intermediate routers and V is the victim machine. The acyclic graph $a_2R_2R_4R_6V$ is said to be the attack path.

Apart from the first straightforward attack scenario, an attacker may use many intermediate compromised hosts (known as stepping-stones) in order to hide his original identity along with launching a distributed attack [6]. The latter issue complicates things even more. We present such an attack scenario in figure 2.

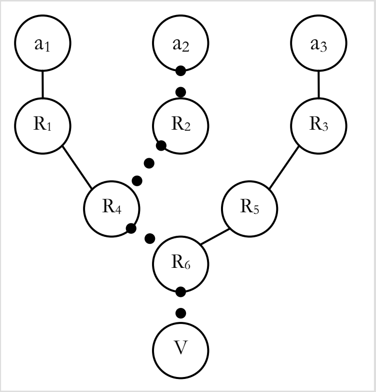

Figure 1 – The attack path.

In this scenario, the attacker uses another intermediate host before reaching to a "zombie" machine. This compromised host, called stepping-stone acts as a conduit for the attacker's communication and is used to change the essence of the attack process. For example, an attacker can use encryption resulting in hiding his actual identity.

The reverse process of an attack (i.e. reconstruction of the attack path back to the originating attacker) that has used one of the previously mentioned masquerade techniques, though, is not straightforward. There are numerous reasons to prevent the correct reconstruction of the attack path (from the victim machine back to the attacker machine). Spoofing is one of the main reasons but existing security functions performed by security countermeasures that are already in place may preclude the capability to follow the reverse path.

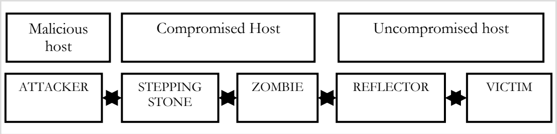

Figure 2 – Attack path using stepping stones [7]

For example, at the network layer, if an attacker lies behind a firewall, even if it could be possible to trace the connection back to the attacker's machine, the firewall would probably block most of traceback efforts (especially those which are based in the ICMP protocol that is often filtered or even blocked). Even if no firewall is in place, most of Internet routers are not allowing for echo-request or echo-reply messages in cross-administrative domains or different autonomous systems. In this case traceback efforts would be fruitless. If an attacker has some of the services of their operating system shut down or has terminated their connection, traceback would stop -at least- one step before them. In addition to this, volatile data that are to be found in operating systems, hubs, routers, switches, firewalls etc. do not often provide state information that could lead to useful conclusions.

Moreover, in the case of a malicious software piece, it is not always possible to trace the original author. This has led to the concept of Software Forensics, originally introduced by Spafford and Weeber in the early-90's [8]. Similar concepts are also to be found from Microsoft Authenticode and Code Signing Extensions introduced by Sun Microsystems Java v.1.1 programming language [9], [10].

Finally, at the system level, temporary files or memory dumps (that contain user activities) are removed after an application-level program has been terminated or the system is restarted. Although there has been a series of techniques developed to hold applications accountable with temporary or removed files that are written from the memory to the system disk as well as disk recovery applications, it is extremely difficult to bind an application with a certain user account (let alone prove that the user account corresponds to the actual owner). The issue of "preservation, identification, extraction, documentation and interpretation of computer data" is mainly addressed by a Computer Forensics analysis [11].

Historically, the concept of Network Forensics (also known as Internet Forensics [12]) was introduced in order to deal with the data found across a network connection (mostly ingress and egress traffic from one host to another). Network Forensics tries to analyze ephemeral data logged through specific security countermeasures (e.g. packet filters, firewalls, intrusion detection systems, etc.) or network devices (e.g. routers, switches etc.). The tools and skills used for a Network Forensics analysis are almost the same used by Internet hackers. It is the ethics and purpose that make the difference. Not surprisingly, Network Forensics evolved as a response to the hacker community [12].

Such an analysis may include Artificial Intelligence and Fusion techniques to speed up the process of gathering and correlating huge audit log files in reaching useful conclusions [13].

3. Tracing a Network Connection

The issue of network tracing is of major importance for network engineers, especially when designing and implementing routing functions and protocols. The Internet Protocol [14] proposes two major options in the IP header that could be used for network tracing: *Record Route* and *Timestamp*, as shown in Figure 3.

The *Record Route* option mandates routing devices along a path to append their addresses to the IP options field. This feature is mostly used for troubleshooting routing issues rather than for accountability issues. Although an IP packet can have a variable length, the total length mandated (in 32-bit words) by the value of the 4-bit IHL (IP Header Length). Given that the maximum value represented by a 4-bit number is 15 (1111 in binary) this results to a maximum length of 60 bytes. Taking under consideration that the fixed size of every IP packet is 20 bytes the maximum size of the IP Options is only 40 bytes (i.e. the size of 10 IP addresses). Considering the current size of the Internet along with the extremely heavy routing information used in current networks the *Record Route* field appears rather limited to withstand recording in every hop a packet traverses.

Secondly, the *Timestamp* option is similar to the *Record Route* option field but every routing device has also to add a 32-bit timestamp for every hop a packet traverses, to be used for debugging routing algorithms [15].

Both these options cannot assist in tracing back a network connection. First of all, there would be a tremendous amount of processing overhead in routing devices, since at least 32-bit information (at least for one hop) has to be appended to data in flight in every routing device. Secondly, since a packet may be routed through different time-zones, there would be the need of a globally synchronized clock for the time-stamps consistency.

Last but not least, a wily attacker can use another option in the IP header options field (e.g. the Loose source routing that mandatory defines a list of routers that should not be missed during routing), "invent" additional hops in the path and fill the 40 bytes available for IP options with false or misleading information. In that case, route recording and time-stamping cannot even be performed.

3.1. IP marking

Having these limitations in mind, three main IP marking (also known as packet marking) approaches have been proposed that enable routers to probabilistically mark packets and therefore reconstruct the complete path [16], [17], [18]. Marking lies to appending data with partial path information so that traceback can be completed. IP Marking approaches use quite complicated mathematical algorithms to identify the origins of sequential IP packets, especially when the source IP addresses are false (i.e. spoofed). So far, IP marking techniques have proved robustness, high probability rates in packet marking and scalable deployment.

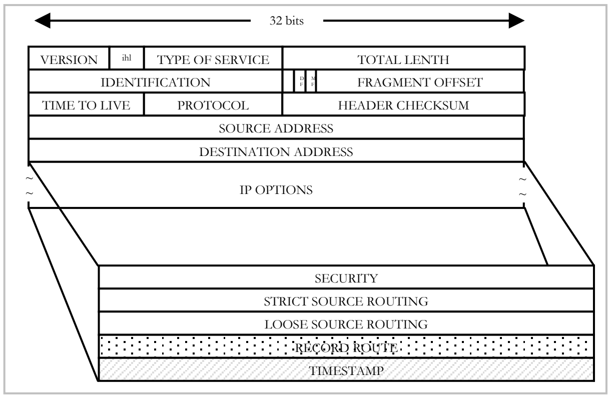

Figure 3 – The Options field in the IPv4 header

On the other hand, marking techniques require that all network traffic is in cleartext while in transit. An obvious issue arising is the compatibility with IPSec [19], especially when operations are performed in the Tunnel mode. In this case the original IP packet is encapsulated in another IP packet and therefore certain portions of the original IP header are cryptographically protected by the Authentication Header of the encapsulating IP packet. Routing devices cannot append marking information in order to achieve traceback. Furthermore, the nature of IP marking aims to reconstruct the edge of the routing path between the attacker and the victim, (i.e. the routing devices that were used) and not in finding the attacker himself. Network and Computer Forensics techniques are also needed in this case to find the actual attacker.

3.2. ICMP traceback

Apart from the IP marking techniques that were discussed above, there are some series of ICMP-based mechanisms evaluated by the research community. Perhaps the most accredited of them is the *iTrace* scheme [20], proposed by Bellovin and currently being the IETF standard. This approach is based upon the capability of routing devices to generate a "trace" packet for every packet they forward and is marked for tracing. At the destination host, the original packet and the “trace” packet are collected and the route is reconstructed. This framework uses HMAC [21] and supports the use of X.509 Digital Certificates for authenticating and evaluating the *iTrace* messages [22]. Under the current *iTrace* proposal, the number of *iTrace* packets generated by a router is small, which implies a low overhead (statistically, around 0.005%) [23]. However, it mainly addresses attacks where a significant amount of traffic comes from a rather small number of sources, due to the lower probability of generating *iTrace* packets [17].

A fair enhancement to the Bellovin’s approach is the Intention-driven *iTrace* schema [23], which is based upon the addition of one extra bit (called intention-bit) in the routing and forwarding process and the functionality provided by the Border Gateway Protocol – BGP [24]. This bit depends on the “usefulness” and “value” of *iTrace* messages. The authors of this approach have also proved the dramatic improvement of *iTrace* in five different DDoS scenarios. Usefulness relies upon the importance an *iTrace* message has. For example, if a host is not under attack (or no attack is detected) or it does not care about tracing then every *iTrace* message is indifferent to that host. On the other hand, “value” reflects the informational value an *iTrace* message has, e.g. *iTrace* messages generated right after the attack has more value that the ones generated after a short period of time.

3.3. Overlay networks

The *CenterTrack* approach is based onto an overlay network by introducing the concept of special types of routers, called tracking routers [25]. Tracking routers have a conceptual (physical or virtual) adjacency with edge routers in an autonomous system. The core of this model is a central tracking system. All edge routers are linked to a central tracking router (or a simple network of tracking routers) via IP tunnels and therefore an overlay network is created. A necessity for the model to perform is that all edge and tracking routers must perform input debugging functions. If no such option is available, the model supports the use of network sniffers for traffic analysis and attack pattern recognition.

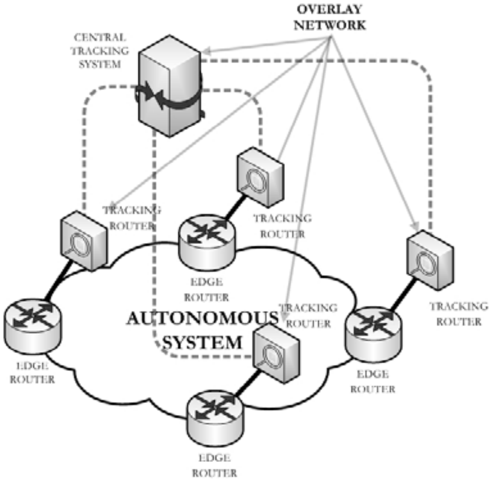

Figure 4 – A simplified version of *CenterTrack*

The malicious traffic destined for the victim is routed through the overlay network via dynamic routing protocols; therefore a hop-by-hop tracking is initiated, starting from the tracking router closest to the victim. Static routes, both in the egress and ingress routers closest to the victim must be configured in a way for attack traffic flows only through the overlay network, allowing at the same time the reception of legitimate traffic. The last function is non-trivial because it is very difficult to filter and reroute only volumes of traffic that match certain attack patterns. In order to succeed in this, *CenterTrack* suggests that an attack pattern matching must be done during the input debugging process either at the edge or the tracking routers.

An important drawback of this model is that it requires application-level intelligence from the (edge and tracking) routers in order to perform pattern recognition and would require more CPU processing power to succeed in this. This feature is an inherent capability of Intrusion Detection Systems (IDS) that are used for similar purposes. At the time of writing, however, the major Internet vendor has announced state-of-the art security features built in edge routers, allowing for signature and anomaly analysis. Furthermore, a wily attacker can detect the presence of tracking systems by statistically measuring the latency via fragmented packets sent to the victim during the information gathering phase of an attack [26]. In addition to this, similar techniques with that used for detection and evasion of IDS systems could be used from an attacker to cause a DoS either to the *CenterTrack* (actually a single-point-of-failure) or the overlay network itself. Finally, if the attack target is the edge router itself then the system would try to reroute traffic destined to the edge router through this specific edge router. This could have either tunnel collapse or routing loops [26].

Baba and Matsuda proposed an alternate mechanism using the concept of an overlay network along with an innovative logging approach [27]. This network is built from sensors that detect attack traffic along with tracing agents (tracers) that log the attack packets and managing agents that coordinate the communication between the sensors and tracers. This approach allows for increased speed as well as less storage requirements by performing selective logging traffic.

3.4. Host-based Identification

Apart from network-based techniques that exploit the functions provided by the Internet Protocol as well as several features of active networking capabilities, there also exist host-based traceback techniques that have been proposed in the first research efforts of the traceback problem. Perhaps the most well-known of them are the *Caller Identification System* (*CIS*) and the *Caller ID* approach that are briefly explained in the following.

The *CIS* is a fully distributed traceback system that aims out identifying the attacker through the login process [28]. The concept of this system relies on the login information exchanges through the systems involved in a connection chain. When a user from host h_1 connects to system h_n (n>2) through intermediate hosts h_2,..h_{n-1} the h_n system recursively queries the h_{n-1} host about the login information. In simple words, for every system where a user requires access all previous login information is checked before access is granted. Apart from being a rather outdated method since it is primarily based on authentication techniques that introduce their own vulnerabilities, it adds an important overhead in the login process so that attackers could be possibly alerted.

Apart from *CIS*, another interesting host-based identification approach has been proposed by Chen [29]. *Caller ID* introduces a manual traceback in every intermediate host of the connection chain. That is, when an attacker connects from h_1 to h_2, h_3, .., h_{n-1}, h_n, the system owner or security personnel break-into hn-1 to verify the origin of the connection, possibly using hacking techniques. He later breaks into h_{n-2} until he reaches h_1 which could potentially be the attacker's machine. Despite the ethics and legal complications of this technique, *Caller ID* does not introduce important overhead like *CIS* and could be scalably augmented to cross-administration domains or even the Internet. The most important limiting factor is the manual processes that have to be performed for every host traced that make this approach rather not-applicable in today's high-speed networks. Despite these limitations, it is said to be used by US Air Force [30].

Commenting on the effectiveness of host-based identification systems these methods are based in trust relationships between the intermediate hosts of the connection chain. That is, if a system is compromised, the attacker can be granted escalated privileges or alter trust relationships between the hosts (e.g. by modifying the hosts file in a Windows system or the .rhosts file in a Unix system), thus presenting his machine as a trusted host. This could result in the lower effectiveness of these methods. Finally, it seems obvious that such a method requires the attack to remain active throughout tracing.

3.5. Application Level

A recent research in automated intrusion response conducted by Network Associates and Boeing Phantom Works has resulted in the Intruder Detection and Isolation Protocol [31], currently being scaled to multiple administration domains across the Internet. The protocol is featuring low cost integration with intrusion detection techniques but is also adding new response mechanisms along with new response algorithms. At the core of the IDIP is the Common Intrusion Specification Language (CISL) was developed by the Common Intrusion Detection Framework (CIDF) as the language providing a unified explanation of a security incident [32]. Recent results have shown that the protocol is performing well when integrated with IDS systems within the DARPA research community [33]. IDIP is running at the application layer of the TCP/IP network model, coordinating intrusion detection and isolation of a security incident.

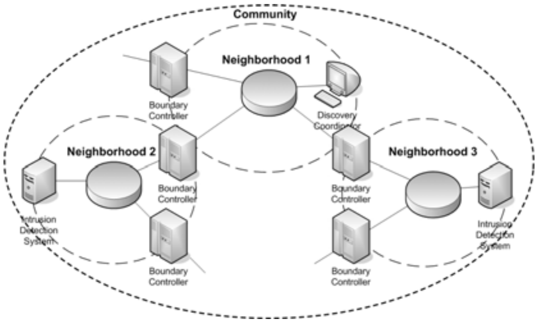

Figure 5– IDIP Communities, Neighborhoods and boundary controllers [33].

The protocol architecture is based upon the systems that belong to the same administrative domain and run the IDIP. These systems, called IDIP components, are forming an IDIP neighborhood, as shown in figure 4. Multiple IDIP neighborhoods, in turn, form an IDIP Community without the need of another coordination component. A component called Discovery Coordination is managing all intrusion detection and response actions within an IDIP Community. Finally, systems running IDIP that belong to more than one IDIP neighborhood are called boundary controllers.

When a connection (or a datagram stream) is in progress within an IDIP-protected network, every IDIP system (node) is auditing the connection for patterns of attack using intrusion detection technologies. When signs of an attack are detected by an IDIP component the detector is informed and, in turn, it spreads the attack information to all the systems within the Community (and further to the IDIP Neighborhood). By this, the attack information is distributed along the path of the attack.

4. 4. Categorization of traceback methods

The different approaches presented in the previous section make the need for a classification a necessity, in order to enhance the power of digital forensics methodologies and counter the limitations of classic Incident Handling & Response capabilities. In the following section we present a multidimensional classification comprising of critical architectural components of each traceback category reviewed. Traceback is one of the most difficult problems in network security and there is a lot of research being conducted in the security world today, therefore a new method is very likely to be proposed from day to day. What we focus on is summarizing their nature, behavior, architecture, applicability and complexity to provide as much feedback as possible to digital forensics methodologies and analyses in order to reach to holding attackers accountable. The results of our classification are presented in Table 1.

4.1. Dimension 1 – Nature

The first important factor is the nature of every traceback method: host-based, network-based or both. We strongly believe that a complete traceback method should empower the characteristics of both worlds. Host-based methods provide accuracy and are more probable to prove the actual attacker while adding more processing overhead. Network-based techniques provide automation, efficiency and effectiveness, while (some of them) are able to detect stepping-stones.

4.2. Dimension 2 - Behavior

The second factor of our classification is the behavior of traceback methods. While some methods perform proactively, by recording connection states or login information, others dynamically correlate ingress and egress traffic. Proactive behavior introduces a significant amount of processing overhead to all tracking devices. On the other hand, reactive behavior greatly reduces processing overhead but is susceptible to misleading information provided by attackers using stepping stones.

4.3. Dimension 3 – Architecture

The third factor of our proposed classification is the architectural model of the traceback solutions. A model of this kind could be either centralized or distributed. A centralized solution, incorporating a central intrusion response module is a single point of failure while providing coordinated responses and decisions. On the other hand, distributed architectural models provide redundancy but suffer time synchronization and response coordination.

4.4. Dimension 4 – Applicability

Our fourth classification factor is the applicability of traceback methods, i.e. how suitable a method is for a given architecture. This field can vary from a private network, an autonomous system, or even the Internet. The more applicable a solution is, the more likely is to contribute to the entire solution of the traceback problem.

4.5. Dimension 5 – Complexity

The last dimension of our classification is the complexity of a traceback method. We define complexity as the amount of re-engineering functions that have to be performed in current Internet infrastructure. The lower the complexity of a method is the more likely is to become more easily adoptable.

5. Future Work

Among our next immediate steps are the evaluation of the proposed methods either in test-beds or, hopefully, in controlled real-world deployments. Apart from technical issues that arise, since most of these methods have produced only prototypes, there are many political issues at this time that prevent us from testing these methods in cross-administrative domains (since cooperation between many ISPs would be required in order to record attack paths or allow for traceback methods within their controlled and protected network infrastructure).

However, we hope to drive heavy international research on this so as to practically evaluate traceback methodologies among the academia or/and the industry, therefore get one step closer to actual attackers.

Furthermore, since there are obvious limitations in nearly-all traceback methods (like political issues, encrypted traffic and stepping stones); a detailed operating framework of traceback mechanisms supporting configurable and user-defined policies (and perhaps security strategies or policies) would provide Network Forensics methodologies a common ground to counter political and legal issues. If traceback methods will develop without proper legal support it is very likely that they will be blamed as privacy intrusion to the actual attacker. This is yet another gigantic step to overcome, apart from false traceback information.

Method	**Nature**	**Behaviour**	**Architecture**	**Applicability**	**Complexity**
IP marking	Network	Proactive	Distributed	Internet	High
ICMP traceback	Network	Proactive	Distributed	Internet	Medium
Overlay Networks	Network	Proactive	Centralized	Autonomous Systems	Medium
Host-based Approaches	Host	Reactive	Centralized	Autonomous Systems, Cross-Administrative Domains	Low
Application Level	Both	Reactive	Centralized	Internet	High

Table 1 – Classification results

6. Summary

The traceback problem is one of the hardest in Information Security and has always been the utmost solution to holding attackers accountable for their actions. This paper presented a brief overview of the traceback problem, while discussed the features of Software, Network and Computer Forensics. In the rest of this paper, various traceback mechanisms were examined and categorized according to their features and modes of operation. Finally, a classification for all traceback methods was proposed in order to assess and combine their benefits so as to provide enough information for Digital Forensics analyses, thus getting one step closer to the actual attacker.

References

[1] Wang, B., Schulzrinne, H., An IP Traceback Mechanism for Reflective DoS Attacks, *In Proceedings of CCECE 2004-CCGEI 2004*, Niagara Falls, May 2004

[2] International Standards Organization, “Code of Practice for Information Security Management”, ISO/IEC 17799:2000

[3] Whalen, S., An Introduction to ARP Spoofing, Revision 1.82, (White Paper), Available at: http://node99.org/projects/arpspoof/arpspoof.pdf, 2001

[4] Bellovin, S.M., Security Problems in the TCP/IP Protocol Suite, *Computer Communication Review, Vol. 19, No. 2, pp. 32-48*, April 1989.

[5] Douligeris, C., Mitrokotsa A., DDoS attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks 44(5): 643-666, 2004*

[6] Zhang, Y., and Paxson, V., Detecting Stepping Stones, *Proceedings of the 9th USENIX Security Symposium, Denver Colorado*, August 14-17, 2000

[7] Lee, S. C. and Shields, C. ,Tracing the Source of Network Attack: A Technical, Legal and Societal Problem, *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, United States Military Academy, West Point, NY, 5-6 June, 2001

[8] Spafford, E. H., Weeber, S. A., Software Forensics: Can We Track Code to its Authors?, Purdue Technical Report CSD–TR 92–010, February 1992

[9] Microsoft Corporation, Introduction to Code Signing, Available:

http://msdn.microsoft.com/library/default.asp?url=/workshop/security/authcode/intro_authenticode.asp, 2005

[10] McGraw, G. & Felten, E.,*Securing Java*, Wiley, London, 1999

[11] Kruse W. and Heiser J., *Computer Forensics*, Addison-Wesley, Canada, 2002

[12] Berghel H., The Discipline of Internet Forensics, *Communications of the ACM, Vol. 46/No.8, August 2003*

[13] Nong, Y. , Giordano, J., Feldman J., and Zhong Q., “Information Fusion Techniques for Network Intrusion Detection”, *IEEE Information Technology Conference, Information Environment for the Future*, Syracuse, NY, USA, September 1998

[14] Postel, J., Internet Protocol, RFC 791, September, 1981, Available at: http://www.ietf.org/

[15] Tanenbaum, A.S., *Computer Networks*, Third Edition, Prentice Hall, NJ, 1998

[16] Savage, S., Wetherall, D., Karlin A., and Anderson, T., Practical Network Support for IP Traceback, Proceedings of SIGCOMM'00, Stockholm, August 28 – September 1 / 2000,Sweden

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

[18] Park, K. and Lee, H., On the Effectiveness of Probabilistic Packet Marking for IP Traceback, *Proceedings of 2001 Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (ACM SIGCOMM)*, August 27-31 / 2000

[19] Kent, S. and Atkinson, R., Security Architecture for the Internet Protocol, RFC 2401, Nov. 1998, available at: http://www.ietf.org/

[20] Bellovin S.M., ICMP Traceback Messages, Internet Draft (work in progress), February 2003

[21] US Department of Commerce, Federal Information Processing Standards Publication 198, The Keyed-Hash Message Authentication Code (HMAC), March 6, 2002

[22] Adams, C., Internet X.509 Public Key Infrastructure Certificate Management Protocols, RFC 2510, Available at:http://www.ietf.org/

[23] Mankin, A., et. al, On Design and Evaluation of Intention-Driven ICMP Traceback, *Proceedings of IEEE International Conference on Computer Communications and Networks*, 2001

[24] Rekhter, Y. and Watson, T.J., A Border Gateway Protocol 4 (BGP-4), RFC 791, Available at: http://www.ietf.org/

[25] Stone, R., CenterTrack: An IP Overlay Network for Tracking DoS Floods, *Proceedings of 9th Usenix Security Symposium*, August 14-17 /2000

[26] McClure, S., Scambray, J., and Kurtz, G., *Hacking Exposed*, McGraw-Hill, London, 2001

[27] Baba, T. and Matsuda, S., “Tracing Network Attacks to Their Sources,” *IEEE Internet Computing, vol. 6, no. 3, 2002*

[28] Jung, H., et al. Caller Identification System in the Internet Environment, *Proceedings of 4th USENIX Security Symposium*, 1993

[29] Staniford-Chen, S., and Heberlein, L. T., Holding Intruders Accountable on the Internet, *Proceedings of IEEE Symposium on Security and Privacy*, 1995

[30] Wang X, et. al., Sleepy Watermark Tracing: An Active Network-Based Intrusion Response Framework, *Proceedings of the 16th international conference on Information security: Trusted information: the new decade challenge*, Paris, June 11 - 13, 2001

[31] Schnackenberg D., Djahandari K., Reid T., and Wilson B., Cooperative Intrusion Traceback and Response Architecture (CITRA), Boeing Phantom Works and NAI Labs, Prepared Under Contract N66001-01-C-8048 for Space and Naval Warfare System Center (SSC), San Diego, February 2002

[32] Feiertag R., Kahn C., Porras P., Schnackenberg D., Staniford-Chen S. and Tung B., “A Common Intrusion Specification Language”, June 1999, Available at http://people.emich.edu/pstephen/other_papers/CISL-Original.PDF

[33] Schnackenberg D., and Djahandari K., “Infrastructure for Intrusion Detection and Response”, *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX)*, Hilton Head, S.C., January 25-27,2000