Network Forensic System for ICMP Attacks

Atul Kant Kaushik
Dept of Electronics and Computer Engg
Indian Institute of Technology Roorkee
Roorkee, India

R. C. Joshi
Dept of Electronics and Computer Engg
Indian Institute of Technology Roorkee
Roorkee, India

ABSTRACT
Network forensics is capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents. This paper addresses the major challenges in collection, examination and analysis processes. We propose a model for collecting network data, identifying suspicious packets, examining protocol features misused and validating the attack. This model has been built with specific reference to security attacks on ICMP protocol. The packet capture file is analyzed for significant ICMP protocol features to mark suspicious packets. The header information encapsulated in the packet capture file is ported to a database. Rule sets designed for various ICMP attacks are queried on the database to calculate various statistical thresholds. This information validates the presence of attacks and will be very useful for the investigation phase. The reduced packet capture size is easy to manage as only marked packets are considered. The protocol features usually manipulated by the attackers is available in database format for next stage analysis and investigation. The model has been tested with a sample attack dataset and the results are satisfactory. The model can be extended to include attacks on other protocols.

Categories and Subject Descriptors
K.4.2 [**Computers and Society**]: Social Issues – *Abuse and crime involving computers;* K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *Unauthorized access (e.g., hacking, phreaking)*

General Term
Security

Keywords
Network forensics, pcap, ICMP, investigation

1. INTRODUCTION

Network security attacks are handled by firewalls and intrusion detection systems. These tools are designed to address prevention, detection, mitigation, and response perspective to an attack. They lack any investigative features as they were not designed with forensics in mind. It is very difficult to trace back the source of attack and prosecute the skillful attackers who are covering their tracks. The analysis, examination and reconstruction of an attack cannot be based on the firewall logs and intrusion detection alerts.

Network forensics is a dedicated investigation technology that enables capture, recording and analysis of network packets and events for investigative purposes. It involves monitoring network traffic and determining if there is an anomaly in the traffic and ascertaining whether it indicates an attack. If it is so then the nature of the attack is also determined. When attacks are successful, forensic techniques enable investigators to catch the attackers. The ultimate goal is to provide sufficient evidence to allow the perpetrator to be prosecuted [1]. The network forensic analysis process involves preparation, collection, preservation, examination, analysis, investigation and presentation phases [2]. The collection, examination and analysis phases are most challenging and difficult. A data reduction technique for Intrusion Detection System (IDS) has been introduced by Lam et al [3]. Similarly there is a need to develop techniques to collect and retain sufficient data for analysis and forensics in a storage efficient manner. Data cannot be selectively collected as important information may be lost. Every data cannot be collected as storage requirements are not infinite. Useful attack information must be collected or extracted from packet captures like protocol types, header information, amount of data transferred, number of packets transferred, length of the link in time, and statistics on specific addresses, protocol flags and login attempts help in forensic processes [4].

The massive volume of packet capture file makes it difficult to store, retrieve, examine and analyze the data. The process also becomes time consuming. The raw data logged to record various network events is stored in the standard Libpcap [5] format. The packets, with specific protocol features used to launch the attacks, can be collected from the packet capture file and marked suspicious. These packets are then gathered for analysis, reducing the file size and making it manageable. The specific protocol packets can also be identified and analyzed.

We propose a network forensic system for ICMP based network attacks which can be extended to any of the network attacks. This model enables forensic experts to analyze the marked suspicious network traffic, thus facilitating cost effective storage and faster analysis of high bandwidth traffic.

We identify the significant features which enable security attacks on ICMP protocol and mark suspicious packets [6]. The header information of protocols in the TCP/IP suite encapsulated in the packet capture file is ported to a database. The protocol attributes of each packet are stored as a record. Rule sets for various ICMP attacks have been designed and are queried on the database to calculate various statistical parameters and thresholds. This information is used for validating the presence of attacks. The packet capture information in database records and related attack data is available for investigation process. This model gives the investigation phase a qualitative data.

The paper is organized as follows: Section 2 provides a background on the network attacks based on the ICMP protocol. It also discusses related frameworks for network forensics based on marking and data reduction. Our proposed 'Network Forensic System' is illustrated in section 3. Significant parameters for various ICMP based network attacks are correlated. Rule sets are designed to identify and generate the statistics for some of the categorized attacks. Section 4 describes the details of the experiments performed and results obtained by applying the proposed model for ICMP based network attacks. Conclusions and future work are presented in section 5.

2. BACKGROUND

2.1 ICMP Protocol

The IP packet travels across the Internet and passes through routers that direct the information and provide simple error handling, for example, when the destination is unreachable. The protocol that performs this operation is called the *'Internet Control and Messaging Protocol'* (ICMP). It provides a mechanism for error handling and general messaging across the IP network layer. ICMP is transported in the payload of the IP packet and has several data structures of its own as defined in [7]. The protocol header is graphically illustrated in Figure 1.

<table>
<tr><td></td><td>Bit 0-7</td><td>Bit 8-15</td><td>Bit 16-23</td><td>Bit 24-31</td></tr>
<tr><td rowspan="5">IP Header (160 bits OR 20 Bytes)</td><td>Version HL</td><td>Type of Service</td><td colspan="2">Length</td></tr>
<tr><td colspan="2">Identification</td><td colspan="2">Flags and Offset</td></tr>
<tr><td>Time To Live (TTL)</td><td>Protocol</td><td colspan="2">Checksum</td></tr>
<tr><td colspan="4">Source IP address</td></tr>
<tr><td colspan="4">Destination IP address</td></tr>
<tr><td rowspan="3">ICMP Payload (64+ bits OR 8+ Bytes)</td><td>Type of Message</td><td>Code</td><td colspan="2">Checksum</td></tr>
<tr><td colspan="4">Quench</td></tr>
<tr><td colspan="4">Data (optional)</td></tr>
</table>

Figure 1. ICMP header

Comer [8] summarizes that "Internet Control Message Protocol allows routers to send error or control messages to other routers or hosts. It also provides communication between the Internet Protocol software on one machine and the Internet Protocol software on another."

The ICMP protocol is used for two types of operations - reporting non-transient error conditions and probing the network with request & reply messages. ICMP messages are

Table. 1 Subset of ICMP Types and Codes

Type	Name	Code
0	Echo Reply	0 No Code
3	Destination Unreachable	0 Net Unreachable 1 Host Unreachable 2 Protocol Unreachable 3 Port Unreachable 4 Fragmentation Needed and Don't Fragment was Set 5 Source Route Failed 6 Destination Network Unknown 7 Destination Host Unknown 8 Source Host Isolated 9 Communication with Destination Network is Administratively Prohibited 10 Communication with Destination Host is Administratively Prohibited 11 Destination Network Unreachable for Type of Service 12 Destination Host Unreachable for Type of Service 13 Communication Administratively Prohibited 14 Host Precedence Violation 15 Precedence cutoff in effect
4	Source Quench	0 No Code
5	Redirect	0 Redirect Datagram for the Network (or subnet) 1 Redirect Datagram for the Host 2 Redirect Datagram for the Type of Service and Network 3 Redirect Datagram for the Type of Service and Host
8	Echo Request	0 No Code
11	Time Exceeded	0 Time to Live exceeded in Transit 1 Fragment Reassembly Time Exceeded
13	Timestamp	0 No code
14	Timestamp Reply	0 No Code
17	Address Mask Request	0 No Code
18	Address Mask Reply	0 No Code
30	Traceroute	
33	IPv6 Where-Are-You	
34	IPv6 I-Am-Here	

therefore classified into two categories: ICMP Error Messages and ICMP Query Messages. Each ICMP message is assigned a number, known as the *message type* which specifies the type of message. Another number represents a *code* for the specified ICMP type. It acts as a sub-type, and its interpretation is dependent upon the message type. A subset of message types and codes are given in Table 1.

2.2 ICMP Attacks

ICMP facilitates sending one-way informational message to a host and informs the source host about errors in datagram processing. These two operations are heavily exploited by the attackers to launch the following attacks:

2.2.1 *ICMP Sweep*

An ICMP sweep [9] is not a direct attack on network, but is definite threat to security. By using a sweep, attackers can determine active hosts and perform more direct targeted attacks specific to those hosts. By sending a series of ICMP 'echo request' packets to every IP on a network segment, an attacker will receive ICMP replies confirming that a host is alive. This process is fairly 'noisy' as the attackers are broadcasting across a whole network range.

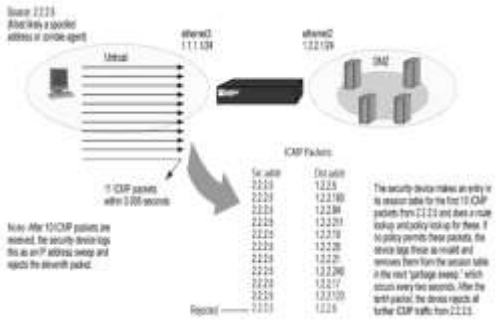

Figure 2. ICMP sweep attack

2.2.2 *Inverse mapping*

Networks are protected by filtering devices such as firewalls and gateways that prevent internal hosts from being reached externally. Attacker uses inverse mapping to obtain a map of an internal network. ICMP reply messages are sent to internal routers about the hosts nearby by and getting the information about the network. This is accomplished without the filtering devices knowing.

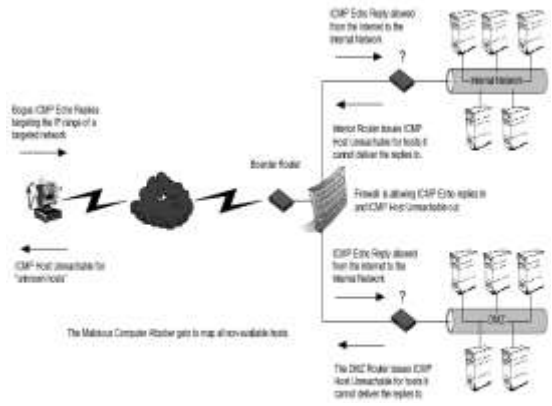

Figure 3. Inverse mapping

2.2.3 *Traceroute network mapping*

Microsoft Windows and all Linux derivatives include a network tool known as traceroute that provides a mechanism for tracking the path of packets flowing between the host and a destination host. It achieves this by utilising the IP protocols TTL (time to live) field, where it attempts to elicit an ICMP 'time exceeded' response from each gateway/router along the path to some host. By default the Linux version of the traceroute application uses UDP to perform its tracing, but it also provides an argument (-I) that allows the tool to use ICMP instead.

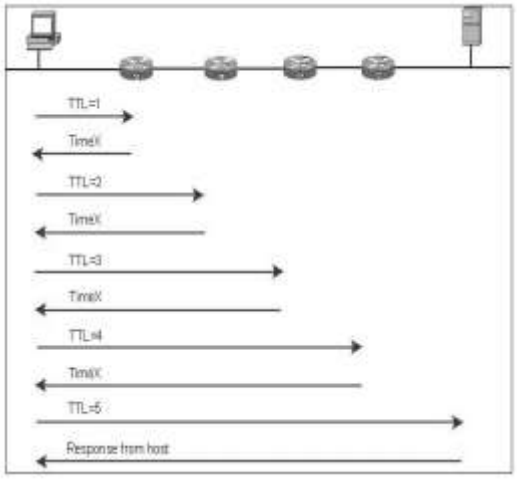

Figure 4. Traceroute network mapping

2.2.4 *OS fingerprinting*

Often an attacker will need to identify what system they are about to attack before they can exploit a vulnerability. In this technique, the attacker relies upon the operating system manufacturer to have built their communications system slightly differently from other operating systems, the steps to recreate this technique are:

- The attacker sends malformed ICMP packets to the destination.
- The destination host will respond with numerous answers to the given requests.
- Each operating system will send slightly different results back to the host. The installed operating system is determined by a process of elimination by evaluating the responses.

This flaw in the development of the operating system, allows specially designed tools to examine the structure of the returned ICMP data and determine the likely operating system.

2.2.5 *ICMP route redirect*

One of the main functions of ICMP is to provide the ability to redirect routing. For example another route has been found to be more efficient or a route failure been discovered. The

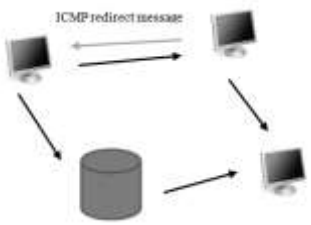

Figure 5. ICMP route redirect

route redirect technique exploits this function by allowing a false ICMP packet to be transmitted telling a target host that they must route information through a new gateway, the 'attacker'. After the traffic is re-routed through the attacker, it can be monitored using a packet sniffing application.

2.2.6 *Ping of death*

The attacker sends excessively large ICMP messages to a target host. Exploiting the weakness in operating system's implementation of the TCP/IP specification, the attacker can

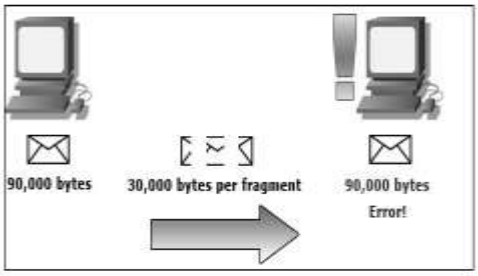

Figure 6: Ping of death

send an ICMP packet greater than the maximum of 65535 octets allowed [10]. The host may become unavailable as a buffer overflow may be created based on the operating system. The computer may crash, force a reboot or make the host hang. A similar attack can be achieved by sending multiple fragmented ICMP packets that requires the operating system to restructure the data on arrival. On examination, the operating system discovers that the packets are not the size they say they are and as a result it forces the machine to hang or reboot.

2.2.7 *ICMP smurf attack*

This attack [11] exploits the weakness in the ICMP and IP protocols by forging the original source address of the packet with the address of the machine to be attack. This 'spoofing' hides the attacker, and begins a chain reaction of network disruption.

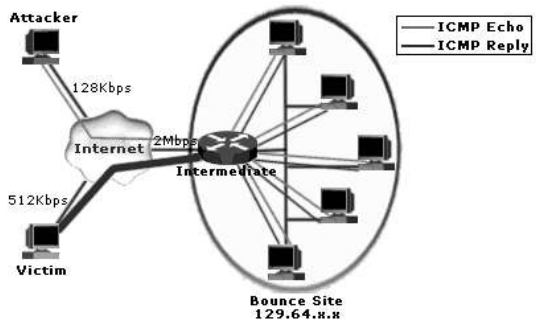

Figure 7. ICMP smurf attack

2.2.8 *ICMP nuke attack*

A Nuke is an old denial-of-service attack against computer networks consisting of fragmented or otherwise invalid ICMP packets sent to the target. It can be achieved by using a modified ping utility to repeatedly send this corrupt data, thus slowing down the affected computer until it comes to a complete stop. Nukes send a packet of information that the target OS can't handle, which causes the system to crash. A specific example of a nuke attack that gained some prominence is the WinNuke, which exploited the vulnerability in the NetBIOS handler in Windows 95. A string of out-of-band data was sent to TCP port 139 of the victim's machine, causing it to lock up and display a Blue Screen of Death (BSOD).

2.2.9 *Attack using source quench*

ICMP source quench messages [12] are generated when a gateway device runs out of buffer space. It is an informational message that is generated in an attempt to inform the remote host generating the traffic to limit the speed at which it is sending network traffic to the remote host. Denial of Service attackers could potentially use ICMP source quench datagrams to rate limit a remote host that listens to unsolicited ICMP source quench datagrams. The attacker needs to guess the sequence number and also match the same with that of the connection. The need to guess the connection source port will lead to many Source Quench messages which do not correspond to an existing connection.

2.3 Packet Capture Format

Libpcap [5] is the library developed by the developers of tcpdump to perform low-level packet (network traffic) capture, read and write capture files and analyze them. This library provides the packet-capture and filtering engines of many open source and commercial network tools, including protocol analyzers / packet sniffers, and network intrusion detection systems. This library also specifies the most basic file format [13] used to save captured network data. The file extension is .pcap. Many applications use the libpcap library as they are able to read the file format. The file has a global header containing some global information followed by zero or more records for each captured packet as shown in Fig 2.

The captured packet in a libpcap file does not contain all the data in the packet as it appeared on the network. It contains at most the first N bytes of each packet. The value of N is called the 'snapshot length'. N will be a value larger than the largest possible packet to ensure that no packet in the capture is sliced. The typical value of N is 65535.

The global header is placed first in the file with fields indicating the file format, byte ordering and the version number. It specifies the correction time in seconds between GMT and the local time zone and the accuracy of time stamps in the capture. The packet capture length N is specified by the field snaplen. The type of data link layer is also mentioned.

The global header is followed by a sequence of packet headers and packet data. The packet header has information fields, ts_sec which gives the date and time when this packet was captured, ts_usec, the microseconds offset to ts_sec

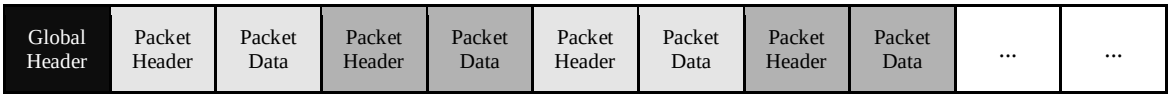

Figure 8. Libpcap packet capture format

when the packet was captured, incl_len the number of bytes of packet data actually captured and saved in the file and orig_len field gives the length of the packet as it appeared on the network. The actual packet data will immediately follow the packet header as a data blob of incl_len bytes without a specific byte alignment.
The libpcap format is very simple and has gained wide usage. It is however limited in not having nanosecond time resolution, inability to display specific connection details, interface information and packet drop count. A next generation format, PCAP NG was proposed [14] and is currently being developed to overcome the deficiencies.

2.4 Related work in network forensics

Identification of significant features to determine attacks is an important challenge in network forensics. Mukkamala and Sung [15] addressed this issue by ranking the importance of input features using two methods namely – performance based method and SVM (Support Vector Machine)-specific feature ranking method. These methods, based on the concept of neural networks, used training time, testing time, and classification accuracy as the performance measure and a set of rules for ranking. The authors presented a description of a few significant features identified for particular types of attacks. However they did not identify the significant features to discover the network attacks.

Almulhem and Traore [6] proposed an architecture of a network forensics system that records data at the host-level and network-level. The main idea was to capture each packet and then mark it as either 'friendly' or 'malicious, using a list of suspicious IP addresses maintained by a group of sensors. The packet was marked if its source IP address was in the list. The sensors were used to update the list as and when it was required and it also maintained some information of each malicious IP. Their proposed system marked the packets based on the list of malicious IP addressed which was assumed to be already present, but it is still an open challenge to identify such a list of IP addresses. The system also did not clearly state the technique used to determine how an IP address can be considered malicious.

Staniford et al. [16] proposed a data reduction approach to infer the event likelihood and only consider the anomalous packets for further analysis. However, the work was only concentrated towards detecting stealthy portscans. Bailey et al. [17] focused on scalable monitoring of darknets and reducing the amount of data for the forensic honeypots by using source-distribution based methods. Maier et al. [18] suggested storing the network traffic up to a cutoff limit of bytes per connection. Our approach, however, focuses on data reduction for forensic analysis of network attacks by correlating the network attacks and corresponding identified significant network features.

3 PROPOSED MODEL FOR NETWORK FORENSICS

We propose a model for network forensic which includes collection of network data, identification of suspicious packets, examining protocol features misused and validation of the attack. This model is built to address the major issue of the large amount of data to be examined for correlation of network features and attacks. This model is elaborated with reference to the network attacks on ICMP protocol. After achieving significant reduction in the network data we validate the system by analyzing the statistics from the database of the protocol header parameters encapsulated in packet captures. The desired results are reported and used for investigation phase. The model is shown in Figure 9 and explained below:

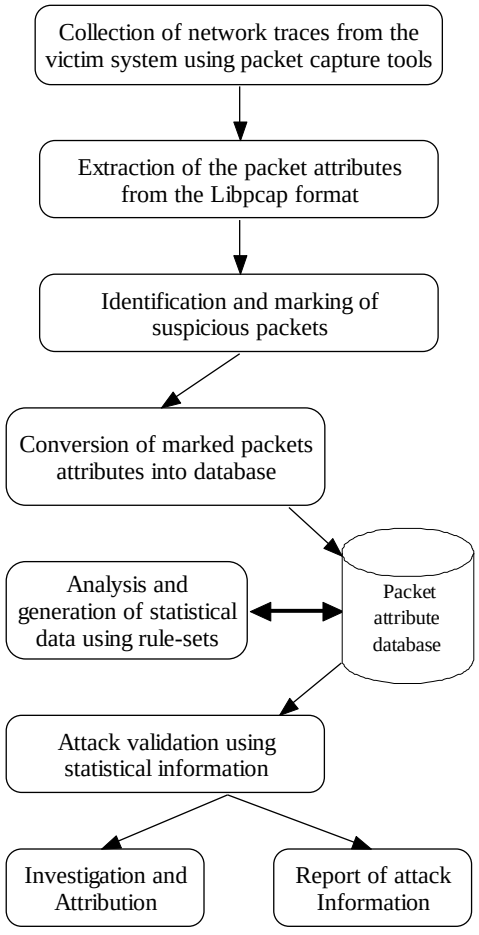

Figure 9. Network forensic system for ICMP attacks

3.1 Collection

The network traces are collected from the victim system using packet capture tools like snort [19], tcpdump [20], wireshark [21] etc. The log file of packets captured results in a very large size in the Libpcap format. The packet attributes are extracted by opening the libpcap file using Perl language modules. The data part of the libpcap file contains the IP packets with information regarding various protocols and their attributes. We select the packets corresponding to ICMP protocol as we focus our model on ICMP attacks.

Table 2: Correlation of attack and protocol features

Attacks on ICMP Protocol	Significant Parameters
ICMP Sweep	Type = 8 and code = 0
Inverse mapping	Type = 0 without sending type = 8
Traceroute network mapping	TTL = 0 and type = 8
OS fingerprinting	Type = 8 and code other than 0
ICMP route redirect	Type = 5
Ping of death	Total size of IP packet > 65535 bytes
ICMP smurf attack	Type = 0 without sending type = 8
ICMP Nuke attack	Invalid packet
Attack using source quench	Type = 4 and code = 0

3.2 Identification and Marking

The significant network parameters corresponding to various network attacks on ICMP protocol are identified and presented in Table 2. This information is used to remove the redundant information which is not relevant for the ICMP based attacks. The traffic packets having these significant attributes violated are marked as suspicious. This reduced set of packets is sufficient to examine the attack packets, analyze them and generate reasonable information about the source of attack.

3.3 Conversion into database

The conversion process creates a database of packet attributes marked from the packet capture file. Each of the marked packets adds a record to the database of all the attributes related to IP and ICMP protocol. We exclude the data portion in both the protocol packets and concentrate only on the header information. 13 attributes of IP and 3 attributes of ICMP protocol are recorded. The timestamp associated with each packet is also recorded. We also store the time and date information available in the packet capture file as a field in the database. These timestamps can be used to convert this database of packet attributes back to the standard packet capture format.

3.4 Analysis

The goal of analysis is to generate statistical data by retrieving information from the packet attribute database. The statistical data is generated using the rule-sets for various attacks on ICMP protocol. The rule-sets store and update various statistical information of the database which helps in the validation of the attack. The attack thresholds are identified from these statistical values. The threshold values are however specific to the network topology and environment. A pseudo code for few of the sample rule sets for ICMP attacks are given below:

3.4.1 ICMP Sweep

```
IF type = 8 and code = 0 THEN
    IF source IP != host IP THEN
            IF source IP is in the database THEN
              update time; scount:= scount+1;
            ELSE record IP,date, time,scount:=1;
    ELSE record destination IP, immark := 1, smark:=
    1, imcount := 0;  smcount := 0;
```

3.4.2 Inverse Mapping

```
IF type = 0 THEN
    IF source IP != host IP THEN
    IF source IP is in the database and immark = 0
    THEN
            Update  time; imcount:=imcount+1;
    ELSE record source IP, date, time;
            imcount := 1;
```

3.4.3 ICMP smurf attack

```
IF type = 0 THEN
    IF source IP != host IP THEN
    IF IP is in the database and smark = 0 then update
          time; smcount:=smcount + 1;
    ELSE record IP, date, time; smcount := 1;
```

3.4.4 Traceroute

```
IF type = 8 and code = 0 and ttl = 0 THEN
    IF source IP != host IP THEN
    IF IP is in the database THEN
            update time; tcount := tcount+1;
    ELSE record IP, date, time;
            tcount := 1;
```

This pseudo-code adds and updates the value of corresponding counts which is used to validate the attacks. The threshold values are compared with this counts which results in the attack information.

The analysis part is not limited to the rule-sets algorithm; other analysis methods based only on the marked suspicious traffics may be the alternative approach. Also the rule-sets presented above are bit different from the rule-sets of the IDS to keep the forensics in mind and to generate the attacker information.

3.5 Validation and Investigation

The statistical information stored in the database is used for attack validation. The analysis of the marked packets produces statistical data which gives closer information about the attacks. The decision about the actual occurrences of the attack and their corresponding source information can be taken by observing the statistics and comparing with the thresholds. We observe that the thresholds should be low for inverse mapping and traceroute attacks and be a little higher for smurf and sweep attack. The reduced database also gives the similar qualitative information about the attacks and their sources as with the original pcap file of large size.

The information report about the different kinds of attacks validated can be used to investigate the actual attacker. This information and the reduced database will help in the investigation of the attacker. Investigation phase involves discovering the attacker by overcoming address spoofing. The analysis module will provide the reduced database having qualitative information that can be directly used for the investigation process. The attacker information reported by our model also helps for investigation of the actual attacker.

4 EXPERIMENTS AND RESULTS

A small network set-up of three hosts, each having the operating system Ubuntu v9.10, is created for the experimentation. Two of the hosts were used to launch ICMP based attacks on the third host. We used wireshark to collect the packet captures in the victim system. The attacking hosts launched ICMP sweep, smurf, OS fingerprinting and traceroute on the victim host. We used nmap [22], sing [23] and traceroute [24] tools to launch attacks.

We created dataset of network traces having ICMP attack data on it. We launched various types of attacks and obtained one file in standard pcap format. Normal internet activity was carried out by the users and this legitimate traffic is also recorded in the file. The size of file was 12.85 MB having 91452 packets. Our proposed model is applied for reduction of the dataset. The total number of marked packets was 5372 which is shown in the figure 10. Most of the ICMP packets were marked as suspicious as expected since most of the packets logged were attack packets and normal activity with ICMP packets were not happening. Database of attributes of the marked packets was created using Perl modules.

We implemented the network forensic system using Perl language. The implementation handles the process of capturing the packets, extracting their content, marking the suspicious packets and creating a database of attributes of the marked packets. We have designed two subroutine modules - *process_the_captured_packets* and *insert_icmp_packets* to accomplish the same. *process_the_captured_packets* subroutine captures the packets, decodes the Ethernet frame and IP datagram, checks if the protocol used is ICMP and accordingly calls the other subroutine. *insert_icmp_packets* subroutine extracts information from the frame, IP datagram, ICMP packet and header. It also marks the packets based on the significant parameters identified and stores the required information in the packet attributes database. SQL queries were used to retrieve and update the information of the packet attribute database. These rule-sets have added the statistical information like thresholds to the database. We are developing the module to achieve the attack validation results. The

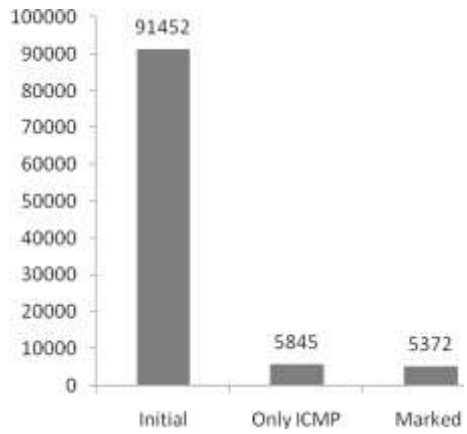

Figure 10. Reduction on Dataset

reduction in data while maintaining the maximum information, will definitely improve the execution time to analyze the large amount of data.

The system will be deployed in one forensic system for network of systems connected in a LAN. The data captured from each of the system will be ported to the forensic system. All the coming data will be integrated and fused to one log file in the forensic system. The proposed system will be executed for the file and the desired results will be collected. The overhead for the LAN is installation of Wireshark in each of the system and installation of the proposed model implementation in the forensic system.

5 CONCLUSION AND FUTURE WORK

The major challenge in network forensics is handling the massive size of network packet capture. It is difficult to store, manage and analyze. We address this problem by reducing the packet capture file size by marking the attack packets using the packet header information only. For marking the attack packets, we correlated various attacks and its corresponding identified significant features. We focused on some specific attacks on ICMP protocol and have tested our approach on a two packet capture files from a victim system.

We would like to extend our work to ensure that the system can handle all the ICMP attacks. We also would like to add some more attacks on various protocols at the network and application layers. The topology and environment effects on the thresholds also need attention. We want to automate the process of rule-sets querying the database. The investigation phase involving various tools and the attack report generated by our model is under development.

6 REFERENCES

[1] Yasinsac, A. and Manzano, Y. 2001. Policies to Enhance Computer and Network Forensics. In IEEE Workshop on Information Assurance and Security.

[2] Ren, W. and Jin, H. 2005. Modeling the network forensics behaviors. In Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks.

[3] K. Y. Lam, L. Hui and S. L. Chung, "A Data Reduction Method for Intrusion Detection", Journal of Systems and Software, 1996.

[4] Shanmugasundaram, K. and Memon, N. 2006. Network Monitoring for Security and Forensics. In proceedings of 2nd International Conference on Information Systems Security.

[5] Jacobson, V., Leres, C. and McCanne, S. Pcap and Libpcap. Lawrence Berkeley National Laboratory, www.tcpdump.org

[6] Almulhem, A. and Traore, I. 2005. Experience with Engineering a Network Forensics System. In Proceedings of International Conference on Information Networking.

[7] Postel, J. Internet Control Message Protocol, RFC 792. http:// tools.ietf.org/html/rfc0792

[8] Comer, D.E. and Stevens, D.L. 1991. Internetworking with TCP/IP.

[9] SANS Institute Reading Room. ICMP attack illustrated. http://www.sans.org/reading_room/whitepapers/threats/icmp_attacks_illustrated_477?show=477.php&cat=threats

[10] Kenney, M. Ping of death. http://www.insecure.org/sploits/ping-o-death.html.

[11] Kumar, S. 2007. Smurf-based Distributed Denial of Service (DDoS) Attack Amplification in Internet. In Proceedings of International Conference on Internet Monitoring and Protection.

[12] Dracinschi, A. and Fdida, S. 2000. Congestion Avoidance for Unicast and Multicast Traffic. In Proceedings of 1st European Conference on Universal Multiservice Networks.

[13] Wireshark, "Libpcap File Format," December, 2008. http://wiki.wireshark.org/Development/LibpcapFileFormat

[14] Degioanni, L., Risso, F. and Varenni, G. 2004. PCAP Next Generation File Format. http://www.winpcap.org/ntar/draft/PCAP-DumpFileFormat.html

[15] S. Mukkamala and A. H. Sung, "Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques", International Journal of Digital Evidence, 2003.

[16] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans", Journal of Computer Security 10, 2002.

[17] Bailey, M., Cooke, E., Jahanian, F., Provos, N., Rosaen, K. and Watson, D. 2005. Data reduction for the scalable automated analysis of distributed darknet traffic. In Proceedings of USENIX/ACM Internet Measurement Conference.

[18] Maier, G., Sommer, R., Dreger, H., Feldmann, A., Paxson, V. and Schneider, F. 2008. Enriching network security analysis with time travel. In Proceedings of ACM SIGCOMM.

[19] A reference manual on Snort Users Manual 2.8.4, http://www.snort.org/assets/120/snort_manual.pdf

[20] Jacobson, V., et al. TCPDump - dump traffic on a network, http://www.tcpdump.org/tcpdump_man.html

[21] Rechard, S. and Warnicke. Wireshark's Users Guide, http://www.wireshark.org/docs/wsug_html_chunked/

[22] Lyon, G., F. Nmap Reference Guide, http://nmap.org/book/man.html

[23] Sing - Send ICMP Nasty Garbage packets to network hosts, http://www.securitydistro.com/toolinfo/31/Sing-MAN-Page.php

[24] Traceroute - print the route packets take to network host, http://www.freebsd.org/cgi/man.cgi?query=traceroute