available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

Network forensic frameworks: Survey and research challenges

*Emmanuel S. Pilli**, *R.C. Joshi*, *Rajdeep Niyogi*

Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand 247667, India

ARTICLE INFO

Article history:
Received 22 December 2009
Received in revised form
13 February 2010
Accepted 16 February 2010

Keywords:
Network forensics
NFATs
Distributed systems
Soft computing
Honeypots
Data fusion
Attribution
Traceback
Incident response

ABSTRACT

Network forensics is the science that deals with capture, recording, and analysis of network traffic for detecting intrusions and investigating them. This paper makes an exhaustive survey of various network forensic frameworks proposed till date. A generic process model for network forensics is proposed which is built on various existing models of digital forensics. Definition, categorization and motivation for network forensics are clearly stated. The functionality of various Network Forensic Analysis Tools (NFATs) and network security monitoring tools, available for forensics examiners is discussed. The specific research gaps existing in implementation frameworks, process models and analysis tools are identified and major challenges are highlighted. The significance of this work is that it presents an overview on network forensics covering tools, process models and framework implementations, which will be very much useful for security practitioners and researchers in exploring this upcoming and young discipline.

1. Introduction

On August 6, 2009, social networking sites like Twitter, Facebook and Google blogger were knocked down by distributed denial of service (DDoS) attacks. Facebook and Google could recover within a day while Twitter staff team worked round the clock in the weekend to deal with the attack as reported in Computer World. Los Angeles Times speculated that perpetrators of the DDoS attack may have been bored teenagers or Russian and Georgian political operatives involved in cyberspace fighting. The newspaper quoted security experts that fingerprints of a sophisticated operation involving botnets were observed and Twitter website had limited capacity to handle incoming traffic. The obvious reason for the success of this attack was that Twitter's network did not have the defenses in place to mitigate a massive DDoS attack. Most traditional security products aren't equipped to handle massive bombardment of packets that happens in a DDoS attack. The lack of solid contingency plan and pro-active security mechanism created a fragile platform vulnerable to attack as reported in ChannelWeb.

Rosenberg referring to the attack on Twitter, wrote that having appropriate tools in place and following correct procedures help to eliminate or mitigate the effects of an attack. A network analysis tool can be used to capture all packets in a common data format for analysis. It can also raise alerts when thresholds are exceeded. Network forensic tools can be used to reconstruct the sequence of events that occur at the time of

* *Corresponding author*. Tel.: +91 1332 285650/5896; fax: +91 1332 273560.
E-mail addresses: emshudec@iitr.ernet.in, emmshub@gmail.com (E.S. Pilli), rcjosfec@iitr.ernet.in (R.C. Joshi), rajdpfec@iitr.ernet.in (R. Niyogi).
1742-2876/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2010.02.003

attack. Crucial information is gained to prevent a similar attack in future even if the present attack could not be prevented.

Habib in his detailed analysis explained that network forensics can be used to analyze how the attack occurred, who was involved in that attack, duration of the exploit, and the methodology used in the attack. It also helps in characterizing zero-day attacks. In addition, network forensics can be used as a tool for monitoring user activity, business transaction analysis and pinpointing the source of intermittent performance issues.

Network forensics is not another term for network security. It is an extended phase of network security as the data for forensic analysis are collected from security products like firewalls and intrusion detection systems. The results of this data analysis are utilized for investigating the attacks. However, there may be certain crimes which do not breach network security policies but may be legally prosecutable. These crimes can be handled only by network forensics (Broucek and Turner, 2001).

Network security protects system against attack while network forensics focuses on recording evidence of the attack. Network security products are generalized and look for possible harmful behaviors. This monitoring is a continuous process and is performed all through the day. But, network forensics involves post mortem investigation of the attack and is initiated *notitia criminis* (after crime notification). It is case specific as each crime scenario is different and the process is time bound.

Network forensics is the science that deals with capture, recording, and analysis of network traffic. The network log data are collected from existing network security products, analyzed for attack characterization and investigated to traceback the perpetuators. This process can bring out deficiencies in security products which can be utilized to guide deployment and improvement of these tools.

Network forensics is a natural extension of computer forensics. Computer forensics was introduced by law enforcement and has many guiding principles from the investigative methodology of judicial system. Computer forensics involves preservation, identification, extraction, documentation, and interpretation of computer data. Network forensics evolved as a response to the hacker community and involves capture, recording, and analysis of network events in order to discover the source of attacks.

In computer forensics, investigator and the hacker being investigated are at two different levels with investigator at an advantage. In network forensics, network investigator and the attacker are at the same skill level. The hacker uses a set of tools to launch the attack and the network forensic specialist uses similar tools to investigate the attack (Berghel, 2003). Network forensic investigator is further at disadvantage as investigation is one of the many jobs he is involved. The hacker has all the time at his disposal and will regularly enhance his skills, motivated by the millions of dollars in stake. The seriousness of what is involved makes network forensics an important research field.

The aim of this work is to provide a detailed overview of network forensics. The paper is organized as follows: definition, categorization and motivation are clearly stated in Section 2. The various tools available for network forensic analysis and security tools which can also be used for specific phases are described in Section 3. Section 4 surveys the existing network forensic models. We use the term 'model' to imply a theoretical representation of phases involved in network forensics. This model may or may not have been implemented. We propose a generic process model for network forensics, considering only the phases applicable to networked environments, based on the existing models.

Section 5 surveys many implementation frameworks of these models. They are discussed under various categories like distributed systems, soft computing, honeypots and aggregation systems. We use the term 'framework' to mean practical implementation. The specific research gaps existing in these framework implementations and major challenges are presented in Section 6. Conclusions and future work are given in Section 7.

2. Background

Network forensics is being researched for a decade but it still seems a very young science and many issues are still not very clear and are ambiguous. The definition, categorization and motivation for this upcoming field are given below.

2.1. *Definition*

The concept of network forensics deals with data found across a network connection mostly ingress and egress traffic from one host to another. Network forensics tries to analyze traffic data logged through firewalls or intrusion detection systems or at network devices like routers and switches.

Network forensics is defined in Palmer (2001) as ''use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities.''

Ranum is credited with defining network forensics as ''capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents.''

Network forensics involves monitoring network traffic and determining if there is an anomaly in the traffic and ascertaining whether it indicates an attack. If an attack is detected, then the nature of the attack is also determined. Network forensic techniques enable investigators to track back the attackers. The ultimate goal is to provide sufficient evidence to allow the perpetrator to be prosecuted (Yasinsac and Manzano, 2001).

2.2. *Classification of Network Forensics Systems*

Network forensic systems are classified into two types each based on various characteristics like purpose, collection and nature:

- Purpose: 'General Network Forensics' to enhance network security and 'Strict Network Forensics' to get evidence

satisfying legal principles and requirements (Ren and Jin, 2005a).

- Collection of Traffic: 'Catch-it-as-you-can' systems where all packets passing through a particular traffic point are captured and analysis is subsequently done requiring large amounts of storage and 'Stop-look-and-listen' systems where each packet is analyzed in memory and certain information is saved for future analysis requiring a faster processor (Garfinkel).
- Nature: The network forensic system is an appliance with hardware and pre-installed software or exclusively a software tool.

2.3. *Motivation for network forensics*

The large number of security incidents affecting many organizations and increasing sophistication of these cyber attacks is the main driving force behind network forensics. The defensive approaches of network security like firewalls and intrusion detection systems can address attacks only from prevention, detection and reaction perspectives. The alternative approach of network forensics becomes important as it involves the investigative component as well (Almulhem and Traore, 2005). Network forensics ensures that attacker spends more time and energy to cover his tracks, making the attack costly. Network criminals will be more cautious to avoid prosecution for their illegal actions. This acts as a deterrent and reduces network crime rate thus improving security.

The attacker is covering his tracks used to cause attacks, making it more difficult to traceback. Internet Service Providers (ISPs) are also being made responsible for what passes over their network (Perry, 2006). Companies doing business on Internet cannot hide a security breach and are now expected to prove the state of their security as a compliance measure for regulatory purposes. The ISO 27001/27002 standard (Information technology – security techniques – information security management) (ISO/IEC 27001, 2005) specifies the requirements for establishing, implementing, operating, monitoring, reviewing, maintaining and improving a documented Information Security Management System (ISMS) within the context of the organization's overall business risks. Comprehensive audit data are to be maintained to meet the compliance requirements of many regulations (netForensics security compliance management).

Sarbanes–Oxley (SOX) Act requires organizations to implement controls over the release of information to individuals or organizations outside the company's network, and implement policies that define how long, and in what manner, electronic communications should be retained. Gramm–

Table 1 – Description of Network Forensic Analysis Tools (NFATs).

Name of the NFAT	Description
NetIntercept	Captures network traffic and stores in pcap format, reassembles individual data streams, analyzes them by parsing to recognize the protocol, detects spoofing and generates a variety of reports from the results
NetWitness	Captures all network traffic and reconstructs the network sessions to the application layer for automated alerting, monitoring, interactive analysis and review.
NetDetector	Captures intrusions, integrates signature-based anomaly detection, reconstructs application sessions and performs multi time-scale analysis on diverse applications and protocols. It has an intuitive management console and full standards based reporting tools. It imports and exports data in a variety of formats.
Iris	Collects network traffic and reassembles it in its native session based format, reconstructs actual text of the session, replays traffic for audit trial of suspicious activity, provides a variety of statistical measurements and has advanced search and filtering mechanism for quick identification of data.
Infinistream	Utilizes intelligent Deep Packet Capture (iDPC) technology and performs real-time or back-in-time analysis. It does high-speed capture of rich packet details, statistical analysis of packet or flow based data and recognizes hundreds of applications. It uses sophisticated indexing and Smart Recording and Data Mining (SRDM) for optimization.
Solera DS 5150	DS 5150 is an appliance for high-speed data capture, complete indexed record of network traffic, filtering, regeneration and playback. DeepSee forensic suite has three softwares – Reports, Sonar and Search – to index, search and reconstruct all network traffic.
OmniPeek	Provides real-time visibility into every part of the network. It has high capture capabilities, centralized console, distributed engines, and expert analysis. Omnipliance is a network recording appliance with a multi-terabyte disk farm and high-speed capture interfaces. OmniEngine software captures and stores network traffic. OmniPeek interface searches and mines captured data for specific information.
SilentRunner	Captures, analyzes and visualizes network activity by uncovering break-in attempts, abnormal usage, misuse and anomalies. It generates an interactive graphical representation of the series of events and correlates actual network traffic. It also plays back and reconstructs security incidents in their exact sequence.
NetworkMiner	Captures network traffic by live sniffing, performs host discovery, reassembles transferred files, identifies rogue hosts and assesses how much data leakage was affected by an attacker.
Xplico	Captures Internet traffic, dissects data at the protocol level, reconstructs and normalizes it for use in manipulators. The manipulators transcode, correlate and aggregate data for analysis and present the results in a visualized form.
PyFlag	An advanced forensic tool to analyze network captures in libpcap format while supporting a number of network protocols. It has the ability to recursively examine data at multiple levels and is ideally suited for network protocols which are typically layered. PyFlag parses the pcap files, extracts the packets and dissects them at low level protocols (IP, TCP or UDP). Related packets are collected into streams using reassembler. These streams are then dissected with higher level protocol dissectors (HTTP, IRC, etc) (Cohen, 2008).

Table 2 – Description of network security and monitoring (NSM) tools.

Name of the NSM tool	Description
TCPDump	A common packet sniffer and analyzer, runs in command line, intercepts and displays packets being transmitted over a network. It captures, displays, and stores all forms of network traffic in a variety of output formats. It will print packet data like timestamp, protocol, source and destination hosts and ports, flags, options, and sequence numbers.
Wireshark	Most popular network protocol analyzer. It can perform live capture in libpcap format, inspect and dissect hundreds of protocols, do offline analysis, and work on multiple platforms. It can read and write files in different file formats of other tools.
TCPFlow	Captures data transmitted as part of TCP connections (flows) and stores it for protocol analysis. It reconstructs actual data streams and stores in a separate file. TCPFlow understands sequence numbers and will correctly reconstruct data streams regardless of retransmissions or out-of-order delivery.
Flow-tools	Library to collect, send, process and generate reports from NetFlow data. Important tools in the suite are – flow capture which collects and stores exported flows from a router, flow-cat concatenates flow files, flow report generates reports for NetFlow data sets, and flow-filter filters flows based on export fields.
NfDump	A suite of tools working with NetFlow format: nfcapd – NetFlow capture daemon reads the NetFlow data from the network and stores it. NfDump – NetFlow dump reads the NetFlow data from these files, displays them and creates statistics of flows, IP addresses, ports etc. nfprofile – NetFlow profiler filters the NetFlow data according to the specified filter sets and stores the filtered data. nfreplay – NetFlow replay sends data over the network to another host.
PADS	PADS is a portable, lightweight and intelligent network sniffer. It is a signature-based detection engine used to passively detect network assets. It can sniff TCP, ARP and ICMP traffic packets. It can move information about unique assets and services seen on the network into permanent storage, output it in CSV or MySQL format or present an user friendly report.
Argus	Processes packets in capture files are live data and generate detailed status reports of the 'flows' detected in the packet stream. The flow reports capture the semantics of every flow with a great deal of data reduction. The audit data are good for network forensics, non-repudiation, detecting very slow scans, and supporting zero-day events.
Nessus	Vulnerability scanner featuring high-speed discovery, configuration auditing, asset profiling, sensitive data discovery and vulnerability analysis
Sebek	Kernel based data capture tool designed to capture all activity on a Honeypot. It records keystrokes of a session that is using encryption, recover files copied with SCP, capture passwords used to log in to remote system, and accomplish many other forensics related tasks.
TCPTrace	Produce different types of output containing information, such as elapsed time, bytes/segments which are sent and received, retransmissions, round trip times, window advertisements, and throughput.
Ntop	Used for traffic measurement, network traffic monitoring, optimization, planning, and detection of network security violations. It provides support for both tracking ongoing attacks and identifying potential security holes including IP spoofing, network cards in promiscuous mode, denial of service attacks, trojan horses and port scan attacks.
TCPStat	Reports network interface statistics like bandwidth, number of packets, packets per second, average packet size, standard deviation of packet size and interface load by monitoring an interface or reading from libpcap file.
IOS NetFlow	Collects and measures IP packet attributes of each packet forwarded through routers or switches, groups similar packets into a flow, to help understand who, what, when, where and how the traffic is flowing. It also detects network anomalies and security vulnerabilities
TCPDstat	Produces per-protocol breakdown of traffic, for a given libpcap file, like number of packets, average rate and its standard deviation, number of unique source and destination address pairs. It is also useful in getting a high-level view of traffic patterns.
Ngrep	A pcap-aware tool that allows specifying extended regular or hexadecimal expressions to match against data payloads. It can debug plaintext protocol interactions to identify and analyze anomalous network communications and to store, read and reprocess pcap dump files while looking for specific data patterns.
TCPXtract	Extracts files from network traffic based on file signatures. It can also be used to intercept files transmitted across networks.
SiLK	Supports efficient capture, storage and analysis of network flow data based on Cisco NetFlow. The tool suite, consisting of collection and analysis tools, provides analysts with the means to understand, query, and summarize both recent and historical traffic data in network flow records. SiLK supports network forensics in identifying artifacts of intrusions, vulnerability exploits, worm behavior, etc. SiLK has performance as a key element and manages the large volume of traffic by storing only the security-related information.
TCPReplay	Suite of tools with ability to classify previously captured traffic as client or server, rewrite layer 2, 3 and 4 headers and finally replay the traffic back onto the network. TCPPrep is a multi-pass pcap file pre-processor which determines packets as client or server, TCPRewrite is a pcap file editor which rewrites packet headers, TCPReplay replays pcap files at arbitrary speeds onto the network and TCPBridge bridges two network segments.
P0f	Passive OS fingerprinting by capturing traffic coming from a host to the network. It can also detect the presence of firewall, use of NAT, existence of a load balancer setup, distance to the remote system and its uptime.
Nmap	Utility for network exploration and security auditing. It supports many types of port scans and can be used as on OS fingerprinting tool. It uses raw IP packets in novel ways to determine hosts available on the network, services being offered, operating systems running, firewalls in use and many other characteristics.

(*continued on next page*)

Table 2 (*continued*).

Name of the NSM tool	Description
Bro	NIDS that passively monitors network traffic for suspicious activity. It detects intrusions by first parsing network traffic to extract its application-level semantics and then executing event-oriented analyzers that compare this activity with patterns deemed troublesome. It is primarily a research platform for IDS, traffic analysis and network forensics.
Snort	Network intrusion prevention/detection system capable of performing packet logging, sniffing and real-time traffic analysis. It can perform protocol analysis, content searching, matching and application-level analysis. It can capture the traffic in libpcap format.

Leach–Bliley Act (GLBA) requires financial institutions to develop, implement, and maintain a comprehensive written information security program that protects the privacy and integrity of customer records.

Health Insurance Portability and Accountability Act (HIPAA) was established to improve health insurance accessibility for individuals changing employers or leaving the workforce and protect the electronic transmission of health-related data. Federal Information Security Management Act (FISMA) recognized the need to define a comprehensive framework for establishing and monitoring security programs for federal agencies. It has minimum security requirements covering 17 security-related areas for protecting the confidentiality, integrity, and availability of federal information and systems.

By adhering to the Payment Card Industry (PCI) Data Security Standard (DSS), retailers, service providers and allied organizations can dramatically reduce the vulnerabilities that are easily exploited for the purpose of compromising corporate data. PCI DSS applies to merchants, acquiring banks, issuing banks, payment processors and other allied service providers that process, store, transmit and/or dispose of consumer card information. All the above mentioned regulatory acts require adherence to strict security measures and maintaining comprehensive audit data. An integrated network forensic process will facilitate meeting compliance requirements for organizations and ISPs, while recording evidence for investigation and helping in understanding the attacker's methodology.

3. Network Forensic Analysis Tools

Network Forensic Analysis Tools (NFATs) (Sira, 2003) allow administrators to monitor networks, gather all information about anomalous traffic, assist in network crime investigation and help in generating a suitable incident response. NFATs also help in analyzing the insider theft and misuse of resources, predict attack targets in near future, perform risk assessment, evaluate network performance, and help to protect intellectual propriety.

NFATs capture the entire network traffic, allow users to analyze network traffic according to their needs and discover significant features about the traffic. NFATs synergize with intrusion detection systems and firewalls and make long term preservation of network traffic records for quick analysis. The attack traffic can be replayed and attackers' moves can be analyzed for malicious intent (Corey et al., 2002). NFATs facilitate organization of captured network traffic packets to be viewed as individual transport layer connections between machines, which enable the user to analyze protocol layers, packet content, retransmitted data, and extract traffic patterns between various machines.

There are some NFATs available that provide reliable data acquisition and powerful analysis capabilities. There are many other open source network security and monitoring tools, which were developed to provide network security. They were not designed with evidence gathering and processing in mind. These tools do not have a forensic standing (Casey, 2004). However they can be used to help in specific activities of forensic analysis. A description about a partial list of the NFATs and network security tools is given in Tables 1 and 2 respectively. The various classifications of tools are visualized in Fig. 1.

3.1. *Commands in modern operating systems*

Many commands are available inbuilt in modern operating systems which can be used for assisting network forensics (Clarke, 2006). A few of them are described below.

- **nslookup** (name server lookup) is a command used to query Domain Name System (DNS) servers to find DNS details and IP addresses of a particular computer.

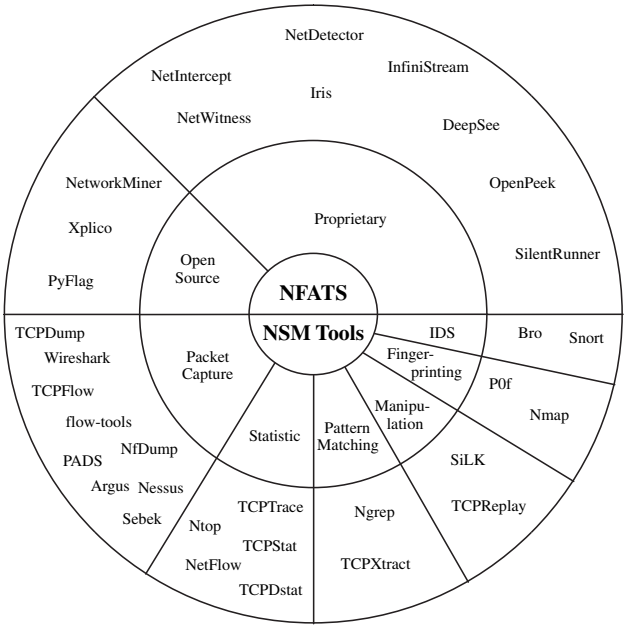

Fig. 1 – Classification of NFATs and NSM tools.

- **traceroute** or **tracert** is a command used to determine the route taken by packets across an IP network.
- **tcpslice** is a program for extracting portions of packet-trace files generated by TCPDump.
- **netstat** (network statistics) is a command tool that displays network connections (incoming/outgoing), routing tables, and other network interface statistics.
- **nbtstat** displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP).
- **whois** is a query/response protocol command used to determine the owner of a domain name, an IP address, or an autonomous system number on the Internet.
- **ping** is a command used to test whether a particular host is reachable across a network.
- **dig** (domain information groper) is a network tool that queries DNS name servers.

4. Network forensic process models

Proven investigative techniques and methods exist for the traditional computer forensic discipline. However as we become more and more networked and mobile in home and business, there is a need to expand our forensic view from disk level to the network level. There is a need to factor this transition into concepts, designs and prototypes. Various digital forensic models were proposed to handle the networked environments since 2001. A generic process model for network forensics is proposed after a brief survey of the existing models.

4.1. *Survey of existing models*

The first attempt to apply digital forensic science to networked environments was taken up as one of the objectives in the first Digital Forensic Research Workshop (DFRWS) (Palmer, 2001) and a framework was proposed. The framework includes the following steps: identification, preservation, collection, examination, analysis, presentation, and decision.

Reith et al. (2002) improvised the above model and produced an abstract digital forensic model that is not dependant on a particular technology or crime. Authors have added preparation and approach strategy phases and included returning the evidence in place of decision. Mandia and Procise (2003) develop an incident response methodology which is simple and accurate. An initial response phase to ascertain the incident and formulation of a response strategy phase are added. The investigation phase includes collection and analysis phases as in the earlier models. Presentation is called reporting and resolution phase suggests improvements, changes, and long term fixes.

Casey and Palmer (2004) proposed an investigative process model to encourage a complete rigorous investigation, ensure proper evidence handling and reduce chance of mistakes. Apart from the common phases, assessment phase validates the incident and a decision is taken whether or not to continue the investigation. Harvesting, reduction, organization and search phases arrange data so that it is the smallest set with high potential evidence. Persuasion and testimony phases present the case in layman terms. Carrier and Spafford (2003) proposed an integrated digital investigation process based on the techniques used for physical investigations. Readiness phase ensures operations infrastructure is ready. Survey, search and collection phases gather and process the data. Reconstruction phase is similar to analysis phase and the documentation phase records all the evidence.

Ciardhuain (2004) combined existing models and proposed an extended model of cyber crime investigations which represents the information flow and performs full investigation. Awareness is the first step which announces investigation. Authorization is taken from internal and external entities. Planning involves strategies and policies. Dissemination is also done for guiding future investigations and procedures.

Baryamureeba and Tushabe (2004) proposed an enhanced digital investigation process model by reorganizing the various phases in Carrier and Spafford (2003). Two new phases, traceback and dynamite are included. Sub-phases like investigation, authorization, reconstruction and communication give clarity and granularity to the major phases. Beebe and Clark (2005) propose a hierarchical, objectives based framework for digital investigative process in contrast to the single tier higher order process models. Their model consists of common phases in first tier. These phases consist of sub-phases, placed in lower tiers, to provide specificity and granularity, guided by principles and objectives.

All models mentioned above are applicable to digital investigation and include network forensics in a generalized form. Ren and Jin (2005a) were the first to propose a general process model for network forensics with following steps: capture, copy, transfer, analysis, investigation and presentation.

4.2. *Proposed generic process model*

We propose a generic process model for network forensic analysis based on various existing digital forensics models discussed above. We formalize a methodology specifically for network based investigation. The various process steps in the model are shown in Fig. 2.

The first phase involves placing of the network security and monitoring tools at strategic places for collecting evidence. The NFATs (NetIntercept; NetWitness; NetDetector; Iris; Infinistream; Solera DS 5150; OmniPeek; SilentRunner; NetworkMiner; Xplico) work in all the other phases, except a few which are not applicable to preservation and investigation phases. PyFlag does not involve packet capture and starts with the examination phase. The network security tools which are applicable to individual phases are also listed in each phase. The various phases in the model are explained below.

4.2.1. *Preparation*

Network forensics is applicable only to environments where network security tools (sensors) like intrusion detection systems, packet analyzers, firewalls, traffic flow measurement software are deployed at various strategic points on the network. The required authorizations and legal warrants are obtained so that privacy is not violated.

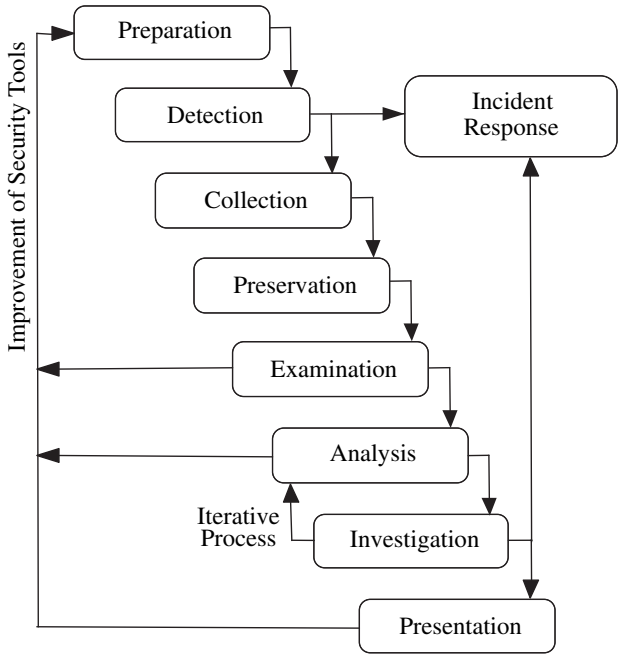

Fig. 2 – Generic process model for network forensics.

4.2.2. *Detection*

The alerts generated by various security tools, indicating a security breach or policy violation, are observed. Unauthorized events and anomalies noticed will be analyzed. The presence and nature of the attack are determined from various parameters. A quick validation is done to assess and confirm the suspected attack. This will facilitate the important decision whether to continue investigation or ignore the alert as a false alarm. Precaution should be taken in order that the evidence is not altered in the process. The tools which assist in this phase are TCPDump, Wireshark, PADS, Sebek, Ntop, P0f, Bro, Snort. This phase branches in two directions - incident response and collection.

4.2.3. *Incident response*

The response to crime or intrusion detected is initiated based on the information gathered to validate and assess the incident. The response initiated depends on the type of attack identified and is guided by organization policy, legal and business constraints. An action plan on how to defend future attacks and recover from the existing damage is initiated. At the same time, the decision whether to continue the investigation and gather more information is also taken. A similar response is to be initiated after the investigation phase (discussed below) where the information obtained may require certain actions to control and mitigate the attack.

4.2.4. *Collection*

Data are acquired from the sensors used to collect the traffic data. A well defined procedure using reliable hardware and software tools, must be in place to gather maximum evidence causing minimum impact to the victim. This phase is very significant as the traffic data change at a rapid pace and it is not possible to generate the same trace at a later time. The amount of data logged will be enormous requiring huge memory space and the system must be able to handle different log data formats appropriately. The tools which assist in this phase are TCPDump, Wireshark, TCPFlow, NfDump, PADS, Sebek, SiLK, TCPReplay, Snort, Bro.

4.2.5. *Preservation*

The original data obtained in the form of traces and logs are stored on a backup device like read only media. A hash of all the trace data is preserved. A copy of the data will be analyzed and the original network traffic data are untouched. This is done to facilitate legal requirements which may expect that the results obtained by the investigation are proved same when the process is repeated on the original data. The tools which assist in this phase are TCPDump, Wireshark, TCPFlow, NfDump, PADS, Sebek, SiLK, TCPReplay, Bro, Snort.

4.2.6. *Examination*

The traces obtained from various security sensors are integrated and fused to form one large data set on which analysis can be performed. There are some issues like redundant information and overlapping time zones which need appropriation. There may be cases where alerts from various sources are contradictory. However this process needs to be done so that crucial information from important sources is not lost. The evidence collected is searched methodically to extract specific indicators of the crime. Minimum attack attributes are identified so that the least information recorded contains the highest probable evidence. A feedback is given to improve the security tools. The tools which assist in this phase are TCPDump, Wireshark, TCPFlow, Flow-tools, NfDump, PADS, Argus, Nessus, Sebek, TCPTrace, Ntop, TCPStat, NetFlow, TCPDstat, Ngrep, TCPXtract, SiLK, TCPReplay, P0f, Nmap, Bro, Snort.

4.2.7. *Analysis*

The indicators are classified and correlated to deduce important observations using the existing attack patterns. Statistical, soft computing and data mining approaches are used to search the data and match attack patterns. Some of the important parameters are related to network connection establishment, DNS queries, packet fragmentation, protocol and operating system fingerprinting. The attack patterns are put together, reconstructed and replayed to understand the intention and methodology of the attacker. A feedback is given to improve the security tools. The tools which assist in this phase are TCPDump, Wireshark, TCPFlow, Flow-tools, NfDump, PADS, Argus, Nessus, Sebek, TCPTrace, Ntop, TCPStat, NetFlow, TCPDstat, Ngrep, TCPXtract, SiLK, TCPReplay, P0f, Nmap, Bro, Snort.

4.2.8. *Investigation*

The goal is to determine the path from a victim network or system through any intermediate systems and communication pathways, back to the point of attack origination. The packet captures and statistics obtained are used for attribution of the attack. This phase may require some additional features from the analysis phase and hence these two phases are iteratively performed to arrive at the conclusion. Attribution is establishing the identity of the attacker and is the most

difficult part of the network forensic process. The two simple strategies of the attacker to hide himself, IP spoofing and stepping stone attack, are still open problems (Guan, 2009). The investigation phase provides data for incident response and prosecution of the attacker.

4.2.9. *Presentation*

The observations are presented in an understandable language for legal personnel while providing explanation of the various procedures used to arrive at the conclusion. The systematic documentation is also included to meet the legal requirements. The conclusions are also presented using visualization so that they can be easily grasped. This process concludes the network forensic analysis as the information presented results in the prosecution of the attacker. The entire case is documented to influence future investigations and to provide feedback to guide the deployment and improvement of security products.

The proposed model is generic as it handles network forensics both in real-time and post attack scenarios. The first five phases (including incident response) handle real-time network traffic. The preparation phase ensures the monitoring tools are in place. Detection phase helps in attack discovery and collection phase captures network packets ensuring integrity of data. A suitable incident response is generated based on the nature of attacks. A hash of the data is created and a copy is made in the preservation phase. The next four phases are common for real-time and post attack scenarios.

The post attack investigation begins at the examination phase, where a copy of the packet capture (libpcap) file is given for investigation. The examination phase fuses inputs from various sources and identifies attack indicators. The analysis phase classifies attack patterns using data mining, soft computing or statistical approaches. The investigation phase involves traceback and attribution. The final presentation phase results in the prosecution of the attacker.

5. Network forensic frameworks

DFRWS proposed the first process model for digital forensics in the networked environments. Many variant models were proposed with different phases as discussed in the previous section. Researchers developed many frameworks which implement these phases and an exhaustive survey is presented category wise to highlight the specific gaps and identify the research challenges.

5.1. *Distributed systems based frameworks*

Internet and LANs are distributed in nature and networks attack events are logged in clients at various locations. There is a need to collect these logs, fuse them and analyze on a central server. A general scheme for the frameworks is shown in Fig. 3. The following implementations were proposed:

Shanmugasundaram et al. (2003) propose ForNet, a distributed network logging mechanism to aid digital forensics over wide area networks. It has two functional components – SynApp, designed to summarize and remember network events for

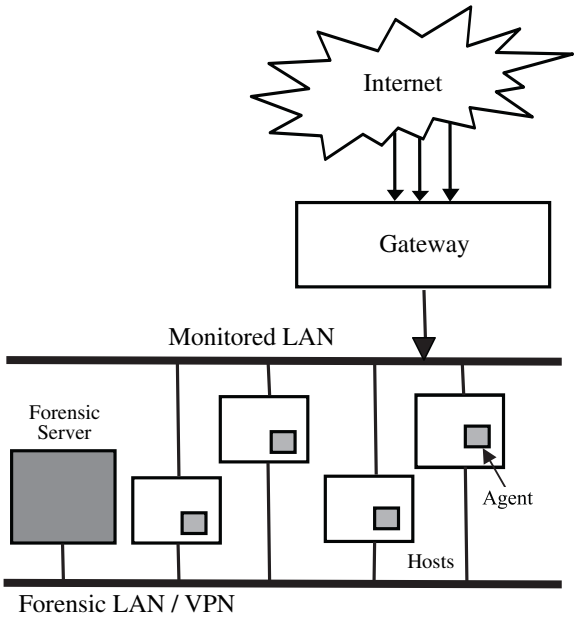

Fig. 3 – A general scheme for distributed frameworks.

a period of time and a Forensic Server, which is a centralized authority for a domain that manages a set of SynApps in that domain. A forensic server receives queries from outside its domain, processes them in co-operation with SynApps and returns query results back to the senders after authentication and certification. The overall architecture involves a network filter, synopsis engine, synopsis controller, configuration manager, security manager, storage management and query processor. Evidence of crimes can be found in packet headers or application dependent payloads. ForNet can identify network events like TCP connection establishment, port scanning, connection record details and uses bloom filters to track other events.

Ren (2004a) proposed a reference model of distributed cooperative network forensic system. It is based on client–server architecture. The server captures network traffic, builds mapping topology database, filters, dumps and transforms the network traffic stream into database values, mines forensic database, and replays network behavior. It also does network surveying, attack statistic analysis and visualization. The distributed agent clients integrate data from firewall, IDS, Honeynet and remote traffic. The goal of this model is dumping misbehavior packets based on adaptive filter rules, analyzing the overall cooperative database to discover the potential misbehavior, and replaying the misbehavior for forensic analysis. It can discover the profile of the attacker and obtain clues for further investigation.

Ren and Jin (2005b) further extended their previous model as distributed agent-based real-time network intrusion forensics system. The goals of this framework include log system information gathering, adaptive capture of network traffic, active response for investigational forensics, integration of forensics data and storing the historical network misuse pattern. The four elements in the system are network forensics server, network forensics-agents, network monitor, and network investigator. Network forensic agents are

engines of the data gathering, data extraction and data secure transportation. Network monitor is a packet capture machine which adaptively captures the network traffic. Network investigator is the network survey machine. Network forensics server integrates the forensics data, analyzes it and launches an investigation program on the network investigator. The model can expedite the investigation of an incident and improve the ability of emergence response.

Tang and Daniels (2005) proposed a simple framework for distributed forensics. It is based on distributed techniques providing an integrated platform for automatic evidence collection and efficient data storage, easy integration of known attribution methods and an attack attribution graph generation mechanism. The model is based on proxy and agent architecture. Agents collect, store, reduce, process and analyze data. Proxies generate the attack attribution graph and perform stepping stone analysis. This model aims at providing a method to collect, store and analyze forensic information. It also provides automatic evidence and quick response to attacks.

Nagesh (2007) implemented a distributed network forensics framework using JADE mobile agent architecture. A node acting as a server, hosting the network forensics-agent, dispatches mobile agents to monitored heterogeneous locations. They gather network traffic logs, examine them and return the results which will be displayed on a user interface. The interface enables the analyst to specify data to be collected and analyze the resultant network events displayed. The solution automates collection of network data from distributed heterogeneous systems using mobile agents. The implementation is scalable, reduces network traffic, addresses single point of failure and provides real-time monitoring.

Wang et al. (2007) developed a dynamical network forensics (DNF) model based on the artificial immune theory and the multi-agent theory. The system provides a real-time method to collect and store data logs simultaneously, provide automatic evidence collection and quick response to network criminals. The system includes a Forensic Server and three agents namely Detector-Agent, Forensics-Agent and Response-Agent. The Detector-Agent captures real-time network data, matches it with intrusion behavior and sends a forensics request message to the Forensics-Agent. The Forensics-Agent collects the digital evidence, creates a digital signature using a hash function and transmits the evidence to the Forensic Server. The Forensic Server analyzes evidence and replays the attack procedure. The Response-Agent is being developed.

The implementations discussed above use client–server/agent–proxy architectures to collect the attack features and analyze them. They are limited in identifying the features correctly and some components are still being developed.

5.2. *Soft computing based frameworks*

The soft computing implementations are used to analyze captured data and classify the attack data. Neural network and Fuzzy tools are used for validation of attack occurrence. A general scheme for the fuzzy logic based frameworks is shown in Fig. 4. The following implementations were proposed:

Kim et al. (2004) develop a fuzzy logic based expert system for network forensics to aid the decision making processes involving sources of imprecision that are non-statistical in nature. The system can analyze computer crime in networked environments and make digital evidences automatically. It can provide analyzed information for a forensic expert and reduce the time and cost of forensic analysis. The framework consists of six components. Traffic analyzer captures network traffic and analyses it using sessionizing. Knowledge base stores rules which are used by the fuzzy inference engine. The rules are written for various attacks using linguistic variables and terms. Membership functions are defined for each fuzzy set and a crisp value of degree of membership is determined. Each input variables crisp value is first fuzzified into linguistic values. Fuzzy inference engine derives output linguistic values using aggregation and composition. Defuzzification defuzzifies the output values into crisp values and the forensic analyzer decides whether the captured packets indicate an attack.

Liu and Feng (2005) proposed the Incremental Fuzzy Decision Tree-Based Network Forensic System (IFDTNFS). This is an efficient way to create a classification model by extracting key features from network traffic by providing the resulting fuzzy decision tree with better noise immunity and increasing applicability in uncertain or inexact contexts. IFDTNFS consists of three components: network traffic separator, traffic detector, and forensic analyzer. The network traffic separator component is responsible for capturing network traffic and separating it according to the service type, and directing the separated traffic to corresponding traffic detector. The traffic detector consists of four components: feature extractor extracts features from network traffic, fuzzy rule base is the knowledge base using which fuzzy decision trees are built, rule base updater adds new samples to the fuzzy decision tree that has been constructed and also adjusts

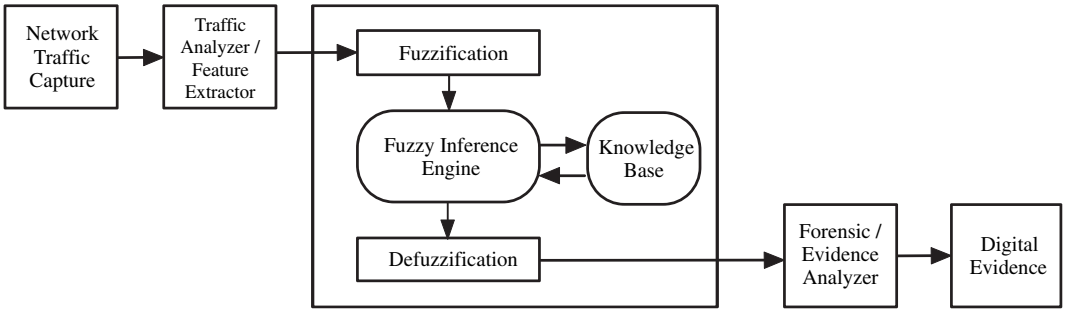

Fig. 4 – A general scheme for fuzzy based frameworks.

the optimal tree size, and fuzzy inferencer fuzzifies input values and processes them with the rule base. Forensic analyzer includes collecting relative event data, analyzing correlated information relating with the event, and establishing digital evidences.

Zhang et al. (2007) propose network forensic computing based on Artificial Neural Network and Principal Component Analysis (ANN–PCA). The major challenge faced in network forensics is massive information to be stored and analyzed. Extraction of key features reduces storage by correlating the features with attacks. ANN–PCA techniques are used to identify all possible violations, extract features and build signatures for new attacks. Classification is done using FAAR algorithm to mine association rules and calculate the PCA values. Classification accuracy increases and information storage size decreases after feature extraction is performed using ANN–PCA.

Neurofuzzy Techniques (Anaya et al., 2009) were used by Anaya et al., to address the challenges of enormous data to be logged and analyzed for network forensic computing. The Neurofuzzy solution is based on Artificial Neural Network (ANN) and fuzzy logic and is used for evidence differentiation into normal and abnormal flows. ANNs are used in information processing to learn from the data and generalize a solution. Fuzzy logic is used to generate a grade of membership to different behaviors so that attacks are determined. The model consists of four modules. The Monitor control module stores all the network information. Information preprocessing module is made up of syntax sub module and correlation sub module. Syntax module is responsible for normalizing the inputs and correlation sub module aggregates the different flow formats and groups the PDUs into a flow. Dependencies module relates all network element logs and takes decision about the flows. The decider module distinguishes between normal and abnormal flows.

Liao et al. (2009) propose a Network Forensic System based Fuzzy Logic and Expert System (NFS-FLES), an effective and automated analysis system, which guarantees evidence reliability by collecting information from different sensors. It also analyzes computer crimes and makes automatic digital evidence using the approach of fuzzy logic and expert systems. The NFS-FLES consists of the following components – traffic capture, feature extractor, fuzzification, fuzzy inference engine, knowledge base, defuzzification and forensic analyzer. The whole operation is done in four parts – real-time forensic data acquisition and preprocessing, knowledge base construction and dynamic rule generation, fuzzy linguistic operation of input attack data and computing aggregation fuzzy value and total fuzzy score of every kind of attack. The forensic result is then output in time.

The soft computing tools give desirable results provided the rules to differentiate attack and legitimate traffic can be generated. Detecting zero-day attacks is also a challenge.

5.3. *Honeypot based frameworks*

Honeypot frameworks are used to attract the attackers so that their process methodology can be observed and analyzed to improve defense mechanisms. The following implementations were proposed:

Honeytraps (Yasinsac and Manzano, 2002) were proposed as a deception tool to collect information about blackhat activities and learn their techniques so that protection and defense mechanisms can be formulated. Honeytraps are Honeypot or Honeynet systems which attract intruders to enter a host by emulating a known vulnerability. Once an attacker penetrates a honeytrap, data are captured to detect and record his actions. This data can be used to profile the tools and tactics used by the attackers putting the investigators in an offensive mode. Two architectures, serial and parallel, facilitate the forensic investigation. The serial architecture places honeytrap between the Internet and the production system. Recognized users are filtered to the production systems and blackhats are contained in the honeytrap. The parallel architecture allows honeytrap to be independent of the production system. Once the system detects the presence of blackhat, the forensic alert system is activated. If the attack is detected, forensic processes are activated on the honeytrap and production systems. Once the attack is contained, the investigation process is begun to determine the identity of the intruder on the production system.

Thonnard and Dacier (2008) propose a framework for attack patterns' discovery in Honeynet data. Their work aims at finding groups of network traces sharing various kinds of highly similar patterns within an attack data set. They design a flexible clustering tool and analyze one specific aspect of the Honeynet data, time series of attacks. Malicious network traffic is obtained from the distributed set of Honeynet responders. Time signature is used as a primary clustering feature and attack patterns are discovered using attack trace similarity. Attacks are detected as a series of connections, zero-day and polymorphic attacks are detected based on similarity to other attacks and knowledge from the Honeynet data is used in intrusion detection efforts. The clustering method does feature selection and extraction, defines a pattern proximity measure and groups similar patterns. The result of clustering applied to time series analysis enables detection of worms and botnets in the traffic collected by Honeypots.

Merkle (2008) investigates automated analysis of network based evidence in response to cyberspace attacks. The two major challenges of network forensics, namely 'complexity' problem of analyzing raw traffic data and 'quantity' problem of amount of data to analyze are addressed in his solution. The model integrates results of data logged by various tools into a single system that exploits computational intelligence to reduce human intervention. This integrated tool is referred as 'automated network forensic' tool. An isolated network of virtual machines is built into a Honeynet. Open source forensic tools are used for collecting data. The information produced by various tools in one stage is characterized and transformed for use by other tools in the succeeding stages. Time consuming and error prone processes are identified and automated. The data sets are partitioned, system is trained and then tested.

Honeynets meet the expectations of forensic analyzers but they cannot be used for investigative purposes as evidence generated is not valid in legal system.

5.4. Attack graph-based framework

Wang and Daniels (2008) developed a graph-based approach toward network forensics analysis. An evidence graph model facilitates evidence presentation and automated reasoning. The basic architecture has six modules: evidence collection module collecting digital evidence from heterogeneous sensors deployed, evidence preprocessing module transforms evidence into standardized format, attack knowledge base provides knowledge of known exploits, assets knowledge base provides knowledge of the networks and hosts under investigation, evidence graph manipulation module generates the evidence graph and attack reasoning module performs semi-automated reasoning based on the evidence graph. A hierarchical reasoning framework consisting of two levels – local reasoning (functional analysis) aims to infer the functional states of network entities from local observations and global reasoning (structural analysis) aims to identify important entities from the graph structure and extract groups of densely correlated participants in the attack scenario. The results from both levels are combined and attacks further analyzed.

5.5. Formal method based framework

Rekhis et al. (2008) develop a system for Digital Forensic in Networking (DigForNet) which is useful to analyze security incidents and explain steps taken by the attackers. DigForNet uses the expertise of intrusion response teams and formal reasoning tools (I-TLA and I-TLC) to reconstruct potential attack scenarios. They integrate analysis performed by the IRT on a compromised system through the use of Incident Response Probabilistic Cognitive Maps (IRPCMs). They provide a formal method framework to identify potential attack scenarios using Investigation-based Temporal Logic of Actions (I-TLA). They generate executable potential attack scenarios and show progress of the attack using Investigation-based Temporal Logic Model Checker (I-TLC), automatic verification tool. Unknown attacks are handled by generating hypothetical actions. The generated executable potential attack scenarios are used to identify risk scenarios that have compromised the system, entities which originated the attacks, different steps taken to conduct the attacks and to confirm the investigation.

5.6. Aggregation framework

Network forensic analysis involves many phases. We have discussed the various security tools in Section 3, which can be used for specific phases. The aggregation frameworks harness the strength of these tools to facilitate forensic investigation rather than building a new tool from scratch. The following implementations were proposed:

Almulhem and Traore (2005) propose a Network Forensics System (NFS) that records data at the host level and network level. The system consists of three main modules – marking, capture and logging. Marking module decides whether a passing packet is malicious. One or more sensors (like IDS) report suspicious IP addresses. Capture module is a collection of lightweight capture modules which wait for the marked packets. They arrange to reliably transport them to the logging module for archival. Logging module is a system repository where the attack data are being stored. It uses three types of loggers – host logger stores data sent by capture module, sensor logger stores sensors' alerts and raw logger is optional and is used when other loggers fail. The capture module was implemented using Sebek, marking module used Snort IDS, and logging module used server-side Sebek, Snort's barnyard tool, ACID Lab and TCPDump.

Nikkel (2006) proposed a Portable Network Forensic Evidence Collector (PNFEC) which was built using inexpensive embedded hardware and open source software. The compact and portable device has been designed for traffic collection between a network and a single node, having specific modes of operation, rapid deployment and stealthy inline operation. The traffic on the Ethernet Bridge is promiscuously captured using various pcap based capture tools and stored on a hard disk. The operating system, additional software, configuration files and investigator activity logs are stored on a compact flash. Administrative access controls various aspects of the device like startup, scheduling, configuration of capturing filters, forensic functions such as preserving and transferring the evidence. The PNFEC is easy to deploy and operate (plug-and-play). The network traffic collected can be stored in encrypted form. PNFEC also controls filtering of captured data using TCPDump to ensure there are no privacy violations. A script is used to create a cryptographic hash of the packet capture files and preserved. OpenBSD is the operating system used, as many of the functionalities like secure access, packet capture, encrypted file system, evidence preservation, disk wiping and formatting tools, are included by default. Tools for trouble shooting (TCPFlow) and pcap management (tcpslice) are also added. PNFEC operates in three modes – investigator, server and user.

Vandenberghe (2008) proposed a Network Traffic Exploration (NTE) Application being developed by Defense Research and Development Canada (DRDC) for security event and packet analysis. This tool combines six key functional areas into a single package. They are intrusion detection (signature and anomaly based), traffic analysis, scripting tools, packet playback, visualization features and impact assessment. NTE has three layers with MATLAB as development environment, low level packet analysis library and unified application front end. It provides an environment where statistical analysis, session analysis and protocol analysis can exchange data.

6. Research challenges

The frameworks and implementations for network forensic analysis have been surveyed in the previous section. The limitations and specific research gaps associated with different phases in each implementation are given below.

6.1. Collection and detection

The first step in network forensic analysis involves collection of network traces and detection of attacks. The traces involve IDS and firewall logs, logs generated by network services and applications, packet captures by sniffers and NFATs (Nikkel,

2005). The challenge is to identify useful network events and record minimum representative attributes for each event so that the least amount of information with highest probable evidence is stored (Mukkamala and Sung, 2003). This results in reduction of data storage requirements. A data digest will be sufficient for discovery of malicious behavior and a full capture is required for reconstruction of attack behavior.

6.2. *Data fusion and examination*

The data captured from various tools must be aggregated and examined to ascertain whether investigation should be commenced. Data fusion of all the logs collected from various security tools deployed in each hosts on the entire network is a crucial problem (Ren, 2004b). The dependencies of packet attributes from various tools and reconnaissance of attributes from different hosts validate an attack. Characterization of anomalous network events and distinguishing attack traffic from legitimate traffic by searching for patterns of anomalies is a major challenge.

6.3. *Analysis*

The critical step in the entire process of network forensics is to analyze attack data and arrive at a conclusion, pointing at the source. Classification and clustering of network events need to be done so that scrutiny of large volumes of data to understand their relationship with attacks becomes easy. Parsing and analysis of complex protocols also needs focus. Pattern recognition of anomalies using soft computing and data mining techniques can be applied for classification, correlation and link analysis. The categorization of attack patterns and attack reconstruction methods used to understand the intention and methodology of the attacker needs research focus (Almulhem, 2009).

6.4. *Investigation*

The investigation must enable attribution of an attack to a host or a network. The results must meet the admissibility criteria in a court of law. The analysis of logs and other network traces must lead to the source of attacks. IP traceback involves tracing back to the source address of the attacker by overcoming IP spoofing (Mitropoulos et al., 2005). Detecting and profiling TCP connection chains can bring out the stepping stones used to launch an attack. Creating a topology database and IP location mapping to locate an attacker geographically is a major challenge. As new protocols like IPv6, become operational and popular, there will be a great need to resolve incidents involving these protocols (Nikkel, 2007).

6.5. *Incident response*

Active real-time response to the network misuse is to be performed so that important data are not lost by the time response is initiated. The response processes are to be launched immediately when alerts begin. The key issue to be maintained is that the attacker must not be aware of the response (Khurana et al., 2009).

7. Conclusions and future work

Network forensics ensures investigation of the attacks by tracing the attack back to the source and attributing the crime to a person, host or a network. It has the ability to predict future attacks by constructing attack patterns from existing traces of intrusion data. The incident response to an attack is much faster. The preparation of authentic evidence, admissible into a legal system, is also facilitated.

We made an extensive survey on various network forensic framework implementations. We also made a study on digital forensic models and propose a generic model. The functionality of NFATs and NSM tools is also discussed. The specific research gaps in these models (existing and proposed) and implementation techniques are identified. The challenges in network forensics as an alternate approach to security are also presented.

We are currently developing a network forensic analysis framework 'Network Forensics System' which is being built using an aggregation of various open source tools. The objective of this framework is to overcome research gaps mentioned above. It will focus on the following phases of our proposed generic framework: traffic collection, detection of attack features, data fusion, examination of network traces, analysis using soft computing and data mining approaches, attack investigation, attribution and incident response.

Future work is to urgently address the limitations and challenges in various tools and framework implementations so that perpetuators of cyber crime are traced back and prosecuted. This will act as a deterrent, resulting in drastic decrease in network crime rate, thereby improving security.

REFERENCES

Almulhem A. Network forensics: notions and challenges. In: Proceedings of the ninth IEEE international symposium on signal processing and information technology (ISSPIT 2009), UAE; Dec. 2009.

Almulhem A, Traore I. Experience with engineering a network forensics system. In: Proceedings of the international conference on information networking (ICOIN 2005), Korea, LNCS 3391. Berlin, Heidelberg: Springer-Verlag; 2005. p. 62–71.

Anaya EA, Nakano-Miyatake M, Meana HMP. Network forensics with neurofuzzy techniques. In: Proceedings of the 52nd IEEE international Midwest symposium on circuits and systems (MWSCAS 2009); 2–5 Aug. 2009, p. 848–52.

Argus, http://www.qosient.com/argus.

Baryamureeba V, Tushabe F. The enhanced digital investigation process model. In: Proceedings of the fourth digital forensic research workshop (DFRWS); 2004.

Beebe NL, Clark JG. A hierarchical, objectives-based framework for the digital investigations process. Digital Investigation (The International Journal of Digital Forensics & Incident Response) Jun 2005;2(2):147–67.

Berghel H. The discipline of Internet forensics. Communications of the ACM 2003;46(8):15–20.

Bro, http://www.bro-ids.org.

Broucek V, Turner P. Forensic computing: developing a conceptual approach for an emerging academic discipline. In: Fifth Australian Security Research Symposium; July 2001.

Carrier, Spafford EH. Getting physical with the digital investigation process. International Journal of Digital Evidence 2003;2(2):1–20.
Casey E. Network traffic as a source of evidence: tool strengths, weaknesses, and future needs. Digital Investigation (The International Journal of Digital Forensics & Incident Response) Feb. 2004;1(1):28–43.
Casey E, Palmer G. The investigative process. In: Casey E, editor. Digital evidence and computer crime. Elsevier Academic Press; 2004.
Ciardhuain SO. An extended model of cybercrime investigations. International Journal of Digital Evidence 2004;3(1).
Clarke GE. Network+ certification study guide. 3rd ed. Osborne McGraw-Hill; 2006. p. 339–89.
Cohen MI. PyFlag – an advanced network forensic platform. Digital Investigation (The International Journal of Digital Forensics & Incident Response) Sept. 2008;5(1):112–20.
Corey V, Peterman C, Shearin S, Greenberg MS, Bokkelen JV. Network forensics analysis. IEEE Internet Computing 2002;6(6): 60–6.
DDOS attackers continue hitting Twitter, Facebook, Google, http://www.computerworld.com/s/article/9136402/.
Denial-of-service got Twitter. Is your network next?, http://news.cnet.com/8301–13846_3–10305241–62.html.
Flow-tools, http://www.splintered.net/sw/flow-tools.
Garfinkel S. Network forensics: tapping the Internet, http://www.oreillynet.com/pub/a/network/2002/04/26/nettap.html.
Guan Y. Network forensics. In: Computer and information Security handbook. Morgan Kaufmann Publishers; 2009. p. 339–47.
Infinistream, http://www.netscout.com/Products/infinistream.asp.
Iris, http://www.eeye.com/Iris.
ISO/IEC 27001. Information technology (security techniques, information security management, requirements), http://www.iso.org/iso/catalogue_detail.htm?csnumber=42103; 2005.
Khurana H, Basney J, Bakht M, Freemon M, Welch V, Butler R. Palantir: a framework for collaborative incident response and investigation. In: Proceedings of the eighth symposium on identity and trust on the Internet, Maryland; April 2009, p. 38–51.
Kim J, Kim M, Noh BN. A fuzzy expert system for network forensics. In: Proceedings of the international conference on computational science applications (ICCSA 2004), LNCS 3043. Springer; 2004. p. 175–82.
Liao N, Tian S, Wang T. Network forensics based on fuzzy logic and expert system. Computer Communications Nov. 2009; 32(17):1881–92.
Liu Z, Feng D. Incremental fuzzy decision tree-based network forensic system. In: Proceedings of the international conference on computational intelligence and security (CIS 2005), LNAI 3802. Springer; 2005. p. 995–1002.
Mandia K, Procise C. Incident response and computer forensics. New York: Osborne McGraw-Hill; 2003.
Merkle LD. Automated network forensics. In: Proceedings of the conference on genetic and evolutionary computation (GECCO 2008); 2008, p. 1929–32
Mitropoulos S, Patsos D, Douligeris C. Network forensics: towards a classification of traceback mechanisms. In: Proceedings of the first international conference on security and privacy for emerging areas in communications networks (SecureComm 2005); Sept. 2005, p. 9–16.
Mukkamala S, Sung AH. Identifying significant features for network forensic analysis using artificial intelligent techniques. International Journal of Digital Evidence 2003;1(4).
Nagesh A. Distributed network forensics using JADE mobile agent framework. Master's thesis. Department of Computing Studies, Arizona State University; 2007, http://www.technology.asu.edu/files/documents/tradeshow/Dec06/asha_nagesh_report.pdf
Nessus, http://www.nessus.org.
NetDetector, http://www.niksun.com.
NetFlow, http://www.cisco.com/web/go/netflow.
Network forensics and digital time travel, http://www.technewsworld.com/story/68651.html.
netForensics security compliance management, http://www.netforensics.com/compliance.
NetIntercept, http://www.sandstorm.net.
NetWitness, http://www.netwitness.com.
NetworkMiner, http://networkminer.sourceforge.net.
NfDump, http://nfdump.sourceforge.net/.
Ngrep, http://ngrep.sourceforge.net.
Nikkel BJ. Generalizing sources of live network evidence. Digital Investigation (The International Journal of Digital Forensics & Incident Response) Sept. 2005;2(3):193–200.
Nikkel BJ. A portable network forensic evidence collector. Digital Investigation (The International Journal of Digital Forensics & Incident Response) Sept. 2006;3(3):127–35.
Nikkel BJ. An introduction to investigating IPv6 networks. Digital Investigation (The International Journal of Digital Forensics & Incident Response) June, 2007;4(2):59–67.
Nmap, http://www.nmap.org.
Ntop, http://www.ntop.org.
OmniPeek, http://www.wildpackets.com.
P0f, http://www.lcamtuf.coredump.cx/p0f.shtml.
PADS, http://passive.sourceforge.net.
Palmer G. A road map for digital forensic research. In: First digital forensic research workshop (DFRWS 2001); 2001, p. 27–30.
Perry S. Network forensics and the inside job. Network Security 2006;2006:11–3.
PyFlag,http://www.pyflag.net
Ranum M. Network flight recorder, http://www.ranum.com/.
Reith M, Carr C, Gunsch G. An examination of digital forensic models. International Journal of Digital Evidence 2002;1: 1–12.
Rekhis S, Krichene J, Boudriga N. DigForNet: digital forensic in networking. In: Proceedings of the IFIP TC-11 23rd international information security conference, IFIP, vol. 278. Springer; Sept. 2008. 637–651.
Ren W, Jin H. Modeling the network forensics behaviors. In: Proceedings of the first international conference on security and privacy for emerging areas in communication networks (SecureComm 2005); Sept. 2005a, p. 1–8.
Ren W, Jin H. Distributed agent-based real time network intrusion forensics system architecture design. In: Proceedings of the IEEE 19th international conference on advanced information networking applications (AINA 2005), p. 177–82.
Ren W. On the reference model of distributed cooperative network forensics system. In: Proceedings of the sixth international conference on information integration and web-based application & services (iiWAS2004), Jakarta, Indonesia; 2004a, p. 771–5.
Ren W. On a network forensics model for information security. In: Proceedings of the third international conference on information systems technology and its applications (ISTA 2004), June 15–17, 2004, Utah, USA, p. 229–34.
Sebek, http://projects.honeynet.org/sebek/.
Shanmugasundaram K, Memon N, Savant A, Bronnimann H. ForNet: a distributed forensics network. In: Proceedings of the second international workshop on mathematical methods models and architectures for computer networks security (MMM–ACNS 2003), LNCS 2776. Springer; 2003. p. 1–16.
SilentRunner, http://www.accessdata.com/silentrunner.html.
SiLK, http://tools.netsa.cert.org/silk/.
Sira R. Network forensics analysis tools: an overview of an emerging technology, GSEC, version 1.4; Jan. 2003.
Snort, http://www.snort.org.
Solera DS 5150, DeepSee, http://www.soleranetworks.com.

Tang Y, Daniels TE. A simple framework for distributed forensics. In: Proceedings of the 25th IEEE international conference on distributed computing systems (ICDCS 2005); June 2005, p.163–9.
TCPDstat, http://staff.washington.edu/dittrich/talks/core02/tools.
TCPDump, http://www.tcpdump.org.
TCPFlow, http://www.circlemud.org/jelson/software/tcpflow.
TCPReplay, http://tcpreplay.synfin.net/trac/.
TCPStat, http://www.frenchfries.net/paul/tcpstat.
TCPTrace, http://www.tcptrace.org.
TCPXtract, http://tcpxtract.sourceforge.net.
Thonnard O, Dacier M. A framework for attack patterns' discovery in honeynet data. Digital Investigation (The International Journal of Digital Forensics & Incident Response) Sept. 2008;5(1):128–39.
Tweets fall silent as Twitter goes down for hours, http://articles.latimes.com/2009/aug/07/business/fi–twitter7.
Vandenberghe G. Network traffic exploration application: a tool to assess, visualize, and analyze network security events. In: Proceedings of the fifth international workshop on visualization for computer security; 2008
Wang W, Daniels TE. A graph based approach towards network forensics analysis. ACM Transactions on Information and System Security (TISSEC) Oct. 2008;12(1).
Wang D, Li T, Liu S, Zhang J, Liu C. Dynamical network forensics based on immune agent. In: Proceedings of the international conference on natural computation (ICNC 2007), vol. 3; Aug. 2007. 651–656.
Why is Twitter so vulnerable to DDoS attack?, http://www.crn.com/security/219300104.
Wireshark, http://www.wireshark.org.
Xplico, http://www.xplico.org.
Yasinsac A, Manzano Y. Policies to enhance computer and network forensics. In: Proceedings of the IEEE workshop on information assurance and security, New York; 2001, p. 289–95.
Yasinsac A, Manzano Y. Honeytraps, a network forensic tool. In: Proceedings of the sixth multi-conference on systemics, cybernetics and informatics, Florida, USA; 2002.
Zhang Y, Ren Y, Wang J, Fang L. Network forensic computing based on ANN–PCA. In: Proceedings of the international conference on computational intelligence and security workshops (CISW 2007); 2007, p. 942–5.