Journal of Network and Computer Applications 66 (2016) 214–235

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Review

# Network forensics: Review, taxonomy, and open challenges

Suleman Khan [a,b,*], Abdullah Gani [a,b,*], Ainuddin Wahid Abdul Wahab [b], Muhammad Shiraz [c], Iftikhar Ahmad [d]

[a] Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya, Kuala Lumpur, Malaysia
[b] Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia
[c] Department of Computer Science, Federal Urdu University of Arts, Science and Technology, Pakistan
[d] Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia

ARTICLE INFO

*Article history:*
Received 8 January 2015
Received in revised form
20 January 2016
Accepted 7 March 2016
Available online 9 March 2016

*Keywords:*
Forensic
Investigation
Cybercrimes
Digital evidence
Intrusion detection

ABSTRACT

In recent years, a number of network forensics techniques have been proposed to investigate the increasing number of cybercrimes. Network forensics techniques assist in tracking internal and external network attacks by focusing on inherent network vulnerabilities and communication mechanisms. However, investigation of cybercrime becomes more challenging when cyber criminals erase the traces in order to avoid detection. Therefore, network forensics techniques employ mechanisms to facilitate investigation by recording every single packet and event that is disseminated into the network. As a result, it allows identification of the origin of the attack through reconstruction of the recorded data. In the current literature, network forensics techniques are studied on the basis of forensic tools, process models and framework implementations. However, a comprehensive study of cybercrime investigation using network forensics frameworks along with a critical review of present network forensics techniques is lacking. In other words, our study is motivated by the diversity of digital evidence and the difficulty of addressing numerous attacks in the network using network forensics techniques. Therefore, this paper reviews the fundamental mechanism of network forensics techniques to determine how network attacks are identified in the network. Through an extensive review of related literature, a thematic taxonomy is proposed for the classification of current network forensics techniques based on its implementation as well as target data sets involved in the conducting of forensic investigations. The critical aspects and significant features of the current network forensics techniques are investigated using qualitative analysis technique. We derive significant parameters from the literature for discussing the similarities and differences in existing network forensics techniques. The parameters include framework nature, mechanism, target dataset, target instance, forensic processing, time of investigation, execution definition, and objective function. Finally, open research challenges are discussed in network forensics to assist researchers in selecting the appropriate domains for further research and obtain ideas for exploring optimal techniques for investigating cyber-crimes.



**Contents**



* Corresponding authors at: Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya, Kuala Lumpur, Malaysia.
*E-mail addresses:* suleman@siswa.um.edu.my (S. Khan), abdullah@um.edu.my (A. Gani).

http://dx.doi.org/10.1016/j.jnca.2016.03.005
1084-8045/

## 1. Introduction

The advent of network forensics envisioned several investigation methods for network security breaches and vulnerabilities. These methods rely heavily on identifying, capturing, discovering, and analyzing network traffic encompassing network devices as well as infrastructure (Nelson et al., 2010). To enable network forensics in the presence of network security suspects, the basic precondition is the ability of an investigator to discover the main objective of the investigation (Raftopoulos and Dimitropoulos, 2013). Usually, there are various types of investigation, i.e. undertake a criminal investigation (Mosa and Jantan, 2013), archive contexture for an internal corporate investigation (Perry, 2006), and retort to a particular network incident (Bejtlich, 2013). Each of these investigations has different intentions and procedures; however, the common goal is to investigate network traffic which is collected during different events at the time of network susceptibilities. These investigations correspond to network attacks (Hoque et al., 2014) and its consequence on the network. In addition, network forensics consents to explore digital evidence in the network traffic after the occurrence of the suspected event (Cusack and Alqahtani, 2013). It allows to reform the complete pattern of network attacks that occurred initially in the network. Traditionally, network forensics reconstructs network attack by capturing network traffic at one device and transmits it to other devices for analysis (Chen et al., 2013a; Ibrahim et al., 2012; Jeong and Lee, 2013; Ren, 2004; Wang et al., 2007). However, it is found that transmitting huge amount of data from one device to another overloads the available communication channel and results in time delays (Ibrahim et al., 2012; Kim and Kim, 2011). Moreover, it affects network forensics performance in real-time situation by resulting in poor incident response. It implies that refined methods are required for analyzing network traffic that can satisfy various requirements of network forensics and network security.

Over the years, an extensive range of network forensics techniques (NFT) has been proposed with distinct investigation approaches (Chen et al., 2013a; Fan and Wang, 2010; Fen et al., 2012b; Lin et al., 2010; Ren and Jin, 2005). Such approaches have the sole goal of extracting legal evidence from network security devices and network communication channels that are connected to such network devices. Legal evidence is analyzed using NFT to identify the origin of the attack. For instance, NFT in Jeong and Lee (2013) shows that by capturing network traffic at router is analyzed to discover the origin of the attack which exposes an intruder. Despite extensive research on NFT, only a single study surveyed network forensics Pilli et al. (2010). The study in Pilli et al. (2010) presents outline on network forensics tools, process model, and framework implementation. A comprehensive study of cybercrime investigation using network forensics frameworks along with a critical review of the current network forensics techniques is lacking. However, no such study has been presented including (Pilli et al., 2010), that focuses on the implementation as well as target datasets of NFT with further illustration on the critical aspect of frameworks for NFT.

This study is conducted in view of the diversity of digital evidence and the difficulty of addressing numerous attacks in the network through NFTs. This paper focuses on NFTs with three distinct objectives: (1) accessibility to network artifacts and infrastructure, (2) allows adequate evidence against intruder, and (3) using NFT as a means to convey information regarding malicious attacks with less false negative results. Two basic reasons motivate this selection: (1) all of these NFT aim to provide digital evidence that require intruders to put in more efforts and time to attack, and (2) this selection narrows down the scope and consents for a comprehensive study of the area. The objective of this study is to provide researchers with a comprehensive insight about the state-of-the-art and open challenges to NFT. This study would be even an important contribution to legislators and security agency committees relating to the formulation of standard legal frameworks.

This paper reviews the fundamental mechanics of NFTs, to determine how network attacks are identified in the network. With reference to an extensive review of related literature, a thematic taxonomy is proposed for the classification of current NFTs based on its implementation as well as target data sets involved in the conducting of forensic investigations. A qualitative analysis technique is presented to explore the critical and significant aspects of existing NFTs. We derive parameters including framework nature, mechanism, target dataset, target instance, forensic processing, time of investigation, execution definition, and objective function for discussing the similarities and differences in current NFTs. Finally, open research challenges are discussed in network forensics to assist researchers in selecting the appropriate domains for further research and obtain ideas for exploring optimal techniques for investigating cybercrimes.

The followings are the contribution of the paper: (a) Classification of the frameworks for NFT on the basis of thematic taxonomy. (b) Analysis of current NFT by discussing the

implications and critical aspects. (c) Comparison of current NFT on the basis of significant parameters for highlighting the similarities and differences. (d) Identification of the challenges in NFT for deploying optimal techniques to investigate cybercrimes.

The paper is organized in the following sections: Section 2 discusses the importance of network forensics. Section 3 proposes the taxonomy of NFTs and explains each technique in terms of its implementation with comprehensive critical review. Section 4 shows a comparison among NFT based on the taxonomy parameters as defined in Section 3.1. Network forensics open challenges are presented in Section 5. Finally, Section 6 presents the conclusive remarks and future directions.

## 2. The importance of network forensics

Emergence of IT industry with its security apprehension is one of the motivational factors in network forensics. Organizations are highly concerned about their network and data security due to many attacks on different companies (Zhu, 2011). In recent years, a number of attacks are performed on different social networks like Google blogger, Facebook, and Twitter (Thapliyal et al., 2013). These social networks are affected by DDoS attack by disgruntled users to crash their function. Moreover, phishing attacks (Layton et al., 2010; Li and Schmitz, 2009) are used to capture confidential information of users, i.e. password of bank account that convert intruders into billionaire within minutes. These criminals have to be traced out and legal evidence is required in the court to convict them. This will make intruders concerned to think twice before performing their attacks in the future. Further, in Perry (2006) discourse that Internet Service Providers (ISP) are responsible for outside network attacks as all network traffic flow passes through these providers. They should restrict malicious packets that act as an attack in the network traffic. Most of the companies perform their business transactions online, which put them at risk due to these security breaches. These cyber-attacks could affect the businesses and cause them to go bankrupt (Smith, 2006).

Both small and large organizations foresee security a backbone entity for their businesses in the current and future era. Federal Information Security Management Act (FISMA) has defined comprehensive security programs for federal agencies (NIST, 2002). Similarly, Health Insurance Portability and Accountability Act (HIPAA) is also following security plans to protect health care data from different malicious attacks (Services, 1996). Companies can attract users towards their market portfolio by providing data and network security in e-transactions, e-business, and other Internet based services by increasing trustworthiness for users and ability to safeguard their interest. Companies have to monitor and analyze their network traffic to detect malicious events and deal with the attack as quickly as possible.

The term malicious in network forensics refers to (1) malicious packets, and (2) malicious traffic program (irregular traffic patterns). Malicious packets can be defined as the packets that violate the principles of network communication, for instance, by exploiting vulnerabilities in security devices including gateways and gaining unauthorized access to the network hosts. Most malicious packets contain forged values in various packet fields, i.e. IP address, TCP flags, and port numbers (Wang et al., 2009). As an example, IP spoofing attack is based on a spoofed IP address in the IP header to enable an intruder to appear as a trusted node and establish connection with a victim node (Zander et al., 2007). Another variation is the land attack in which an intruder inserts the source and destination IP addresses of the victim. This causes the victim to get into a loop of self-connection attempts. TCP flags can be altered in the packet to indicate various events including starting and ending of a connection, highest priority of the data, and pushing of the data. That is, a SYN TCP flag indicates the starting of the connection while a FIN TCP flag indicates the end of the previous connection. An attacker can use a combination of both flags, i.e. SYN-FIN to avoid detection at security devices i.e. firewall and intrusion detection systems (Noh et al., 2003). Most intrusion detection systems look for SYN and FIN flags to know about the start and end of a connection. Therefore, any alteration performed in the field of TCP flag is considered to be malicious. Attackers can also change the source and destination port numbers of a packet to perform abnormal communication. For example, if an attacker assigns the same value to the source and destination port then the packet is discarded at the network device. Besides packet tempering, fragmentation of the packet also acts as a source of the attack to make the communication suspicious (Anderson, 2001). For instance, when an IP packet is too large to be transmitted, it is fragmented. In tiny fragment attack, packet is divided into two TCP fragments such that one fragment contains small information including port numbers while the rest of the information is inserted into the other fragments. Network devices are unable to detect such tiny fragments which results in bypassing various network devices. Similarly, large fragment of packets are used to crash the system by affecting the reassembly process at receiving device. For example, in Ping of Death attack, an echo request message is sent to the host with a size larger than the normal packet size (Sperotto et al., 2010).

On the other hand, malicious program refers to irregular traffic flows than usual. For instance, the request of HTTP GET to Amazon's web services is not treated as malicious unless it intends to alter and communicate with command and control (C&C) part of the infrastructure (Perdisci et al., 2010). Malicious programs are compiled unsolicited and stored unauthorized (Hsiao et al., 2010). Generally, malicious programs are divided into three categories based upon their characteristics: (a) Virus, (b) Worm, and (c) Trojan horse. Virus is a malicious program that does not execute independently but it attaches itself to other programs and replicates itself. Worm is a self-replicated malicious program that replicates itself in an active memory without altering other files. Trojan horse is a malicious program that does not replicate like a virus or a worm, however, it affects the network nodes through fraudulence, i.e. email attachments, downloads, backdoor, spyware, and others.

In order to detect malicious packets or malicious programs, active monitoring of certain events is carried out. Techniques for active monitoring include anomaly detection (Chandola et al., 2009), signature scan detection (Li et al., 2008), intrusion detection systems (Jiang et al., 2000; Liao et al., 2013), access control list (Yu and Wang, 2013), and honeypots (Li and Schmitz, 2009). Anomaly detection techniques are used to create patterns of the user's behavior and resources of the network. Any behavior that diverts from these patterns (irregular traffic patterns) is considered malicious (Chandola et al., 2009). Signature scan technique is used to store the signature of the traffic in a database. A passive scan is performed on the network traffic (Li et al., 2008) for irregular pattern recognition. If the pattern matches with the stored signature, it is considered malicious. It is usually performed for known attacks in the network. Intrusion detection systems usually detect intrusion based on pattern matching and statistical anomaly (Igure and Williams, 2008). However, statistical anomaly identifies malicious traffic based on deviations from the normal usage pattern. The normal usage pattern is established initially for determining a deviation in the network traffic. Access control list is used to detect malicious traffic by matching packet headers with pre-defined rules (Yu and Wang, 2013). Honeypot is a trap placed to monitor and prevent intruders from entering into protected area of the network (Li et al., 2008). Honeypot is a disguise that replicates to protect server and induces intruders to interact.

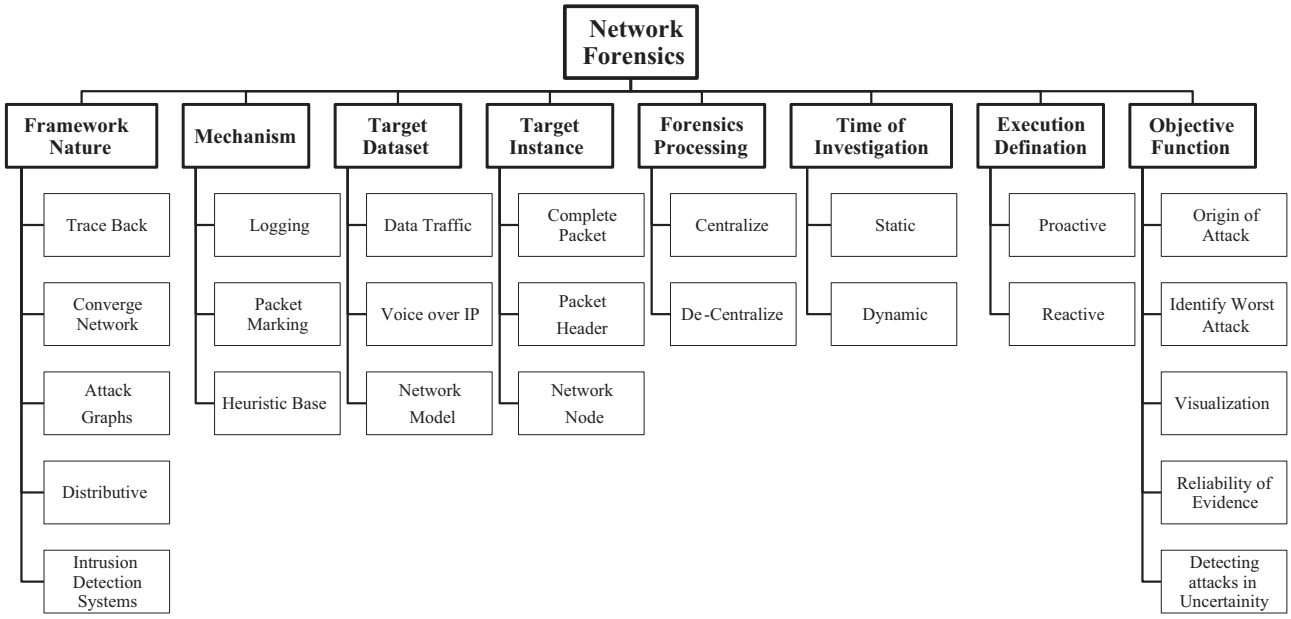


**Fig. 1.** Thematic taxonomy of network forensics.

Subsequently, an attack is detected by monitoring unauthorized probing of open ports in the honeypot.

Network forensics analyzes historical network data in order to investigate security attacks by reconstructing sequence of security attacks (Chen et al., 2013a). It further allows rebuilding of messages, FTP traffic, emails, and other communications from captured and collected network packets. Beside network attacks, network forensics is applicable to address network issues of business critical systems, i.e. finding device to generate attacks, stop virus, increase network performance, and monitor network flow for corporate defiance on a daily basis.

**Table 1**
List of acronyms.

| Symbols | Description |
|---|---|
| AEMS | Authenticated evidence marking scheme |
| AGFE | Attack graph for forensic examination |
| AGVI | Attack graph visualization and interaction |
| AIDF | Analytical intrusion detection framework |
| APIDS | Adaptive intrusion detection system |
| BecIDS | BN event classification intrusion detection system |
| CBSH | Cost-benefit security hardening |
| CFS | Collaborative forensics scheme |
| DCNFM | Distributed cooperative network forensics model |
| DFITM | Dynamic forensics intrusion tolerance method |
| DNF-IA | Dynamical network forensics-immune agent |
| DRNIFS | Distributed real time network intrusion forensics system |
| FCM | Fuzzy cognitive map |
| FS-NIDS | Feature selection-network intrusion detection system |
| HB-SST | Hopping based spread spectrum techniques |
| IDS | Intrusion detection system |
| IIFDH | Intrusion investigation framework data hiding |
| ITP | IP traceback protocol |
| LWIP | Lightweight IP traceback |
| MLL-AT | Multi-level and layer attack tree |
| NFAT | Network forensic analysis tool |
| NFEA | Network forensics evidence acquisition |
| NFIDA | Network forensics based on intrusion detection analysis |
| NFTs | Network forensics techniques |
| NO | Network operators |
| PBNF | Pattern based network forensics |
| SA | Scalable analysis |
| SIP | Session initiation protocol |
| TDPM | Topology assisted deterministic packet marking |
| TTL | Time-to-live |
| VoIP | Voice over internet protocol |
| VoIPEM | Voice over internet protocol evidence model |
| VoIP-NFDE | VoIP-network forensics with digital evidence |

## 3. Current network forensics techniques

This section proposes taxonomy of NFT, its implementation evaluation, and a critical review. The proposed taxonomy is based on the implementation and target datasets of current NFT. Each NFT is reviewed focusing on its implications and critical aspects Table 1.

### 3.1. Thematic taxonomy of network forensics techniques

Fig. 1 shows the proposed thematic taxonomy for the classification of current NFT. The taxonomy contains different parameters which includes framework nature, mechanism, target dataset, target instance, forensic processing, time of investigation, execution definition, and objective function. *Framework nature* parameter shows the primary mechanism of the implementation of NFT. Different framework natures include traceback, converge network, attack graphs, distributive, and intrusion detection system (IDS). *Mechanism* parameter shows the method used by NFT to perform forensic analysis on logs, packets, and network events. Such methods include logging, packet marking, and heuristic based approach. *Target dataset* parameter identifies scope of the traffic on which network forensics is performed. Target datasets include data traffic, voice over IP, and network model. *Target instance* parameter shows instance of the traffic that is captured for network forensics investigation including complete packet, packet header, and network nodes. *Forensic processing* parameter determines the location of forensic investigation which is either centralized or decentralized. *Time of investigation* parameter identifies whether network forensics is performed statically or dynamically. *Execution definition* parameter identifies the type of approach used by NFT i.e. proactive and reactive. *Objective function* parameter explains the aim for which NFT is being performed. For instance origin of attack (Jeong and Lee, 2013), visualizing the attacks (Harbort et al., 2011), reconstructing the attacks (Ibrahim et al., 2012), forensic explanation (Sy, 2009), dynamic forensics

(Chen et al., 2009), reliability of evidence (Fan and Wang, 2010), analyze intrusion data (Jiang et al., 2012), scalable and impact analysis (Albanese et al., 2011), identifying multi-stage network attacks (Fen et al., 2012b), evidence collection (Wang et al., 2007), identifying worst attacks (Diamah et al., 2012), cost-benefit security harden (Zhang et al., 2012), event classification (Kruegel et al., 2003), evidence reduction (Cheng et al., 2013), signature recognition (Jemili et al., 2007), prevention of novel attacks (Boruah and Hazarika, 2014), and effective feature selection (Zhang and Wang, 2013).

### 3.2. Review of network forensics techniques

This section review frameworks of current NFT. Current NFT are qualitatively analyzed by highlighting the implication and its critical aspect. At the end of each sub-section below, a brief table is provided which illustrates each NFT in terms of its objective, forensic approach, methodology, detection of attack, characteristics, and performance. The *objective* attribute highlights the aim of current NFT for which it has been proposed. The *forensic approach* attribute depicts the technique used by NFT while *methodology* attribute shows the method used to analyze network attacks. The *detection of attack* attribute shows the attack investigated by proposed technique is known in advance or not. The *characteristics* attribute indicates different features of the proposed technique while *performance* attribute demonstrates the output based on the proposed solution. Each table (Tables 2–6) provides insight to each NFT for in-depth explanation and understandability.

#### 3.2.1. Traceback based NFT

The identification of the origin of packets in a network is called traceback or IP traceback (Khan et al., 2014b). It is used to identify the source from where packets are generated by identifying the origin of an attack (Akyuz and Sogukpinar, 2009). Traceback is an appropriate NFT used to identify origin of packets by investigating attack path especially for DDoS and IP spoofing attacks (Li et al., 2004). Traceback is more significant due to botnet (Mizoguchi et al., 2011) and DDoS attacks (Chen et al., 2007; Dou et al., 2012) that are witnessed in different distributed networks. Distributed network systems which collaborates with Internet provide seemly atmosphere and attract bot-master for attacks (Kok and Kurz, 2011). To overcome these attacks, it is necessary to keep the network system secured by incorporating various traceback mechanisms in an efficient manner. Various traceback NFT are proposed which are briefly discussed below and also illustrated in Table 2.

Topology assisted Deterministic Packet Marking (TDPM) technique (Wang and Wang, 2010) is proposed for IP traceback to work against Denial Of Service (DoS) and DDoS attacks. In Jin and Yang (2006), the proposed traceback technique use fragments of IP address to traceback origin of the attack by performing hash coding at the victim node. However, the proposed method (TDPM) use fragments of hash, determine initially for IP address to traceback attacks in the network. The router use hash values inside the packet to further compare it at the victim node. The victim node identifies ingress router that use to generate malicious packets by using breadth first search. Once the ingress router is identified, receive hash fragments from other routers at victim node are compared with ingress router hash fragments to investigate malicious packets. It identifies source of the attack by tracking routers from where malicious packet pass through. The advantage of proposed method (Wang and Wang, 2010) includes less hash fragments needed to traceback source of the attack however, other techniques (Belenky and Ansari, 2003, 2007; Jin and Yang, 2006; Yaar et al., 2005) requires massive amount of fragments to traceback source of the attack. However, TDPM is time consuming and

**Table 2**
Summary of traceback network forensics techniques.

| Traceback | Objective | Forensic approach | Methodology | Detection of attack | Characteristics | Performance |
|---|---|---|---|---|---|---|
| TDPM | Traceback DoS attacks | Hash correction codes | Packet marking | Known attacks | Accuracy and scalability | Less false positive rates |
| CFS | Determine fake values in SIP request | Collaborative scheme | Network operator records | Unknown attacks | Time and storage efficient | Less time consuming in analysis |
| NFEA | Provides integrity to the collected evidence | Authenticated evidence and flow-based selection marking scheme | Packet marking | Known attacks | Minimize overhead of network throughput | Performance enhance when applied to selected packets |
| LWIP | Trace DDoS attacks | Lightweight IP traceback | Packet marking | Known attacks | Path reconstruction | Significant path reconstruction |
| Scalable-NF | Self-propagating attacks identification | Scalable based network forensics | Logging | Known attacks | Accuracy, space and time efficient | Capture real timetraffic |
| HB-SST | Identify attack in anonymous communication | Hopping based spread spectrum | Spread spectrum | Unknown attacks | Accuracy and secrecy | False positive rate decreases |
| ITP | Real-time attack investigation | Real-time and periodic analysis | Logging, packet marking | Unknown attacks | Space efficient probabilistic data structure | Accurate attack detection, less false positive rate |

**Table 3**
Summary of converge network forensics techniques.

| Converge network | Objective | Forensic approach | Methodology | Detection of attack | Characteristics | Performance |
|---|---|---|---|---|---|---|
| PBNF | Identification of attack patterns | Log correlation | Logging | Known attacks | Robust and flexible | Reduce false positive rate |
| VoIP-NFDE | Identification of malicious packet in network traffic | Digital evidence with network forensics | Logging | Known attacks | Accuracy, storage efficient | Filtering of network traffic for analysis |
| VoIPEM | Reconstructs attack path | VoIP evidence model | Logging | Known attacks/ Unknown attacks | Integrity and reliability | Identification of attacks within less information |

**Table 4**
Summary of attack graph network forensics techniques.

| Attack graph | Objective | Forensic approach | Methodology | Detection of attack | Characteristics | Performance |
|---|---|---|---|---|---|---|
| SA | Identify attack and their impact on enterprises | Scalable analysis | Dependency graph | Known attacks/ Unknown attacks | Measure current and future attacks | Efficient for small network |
| AGFE | Monitor intruder actions | Anti-forensics | Forensics examination | Known attacks | High overall security, accuracy | Evaluate the alteration in traces |
| MLL-AT | Investigation of multi-level attacks | Multi-level and layer attack tree | Network modeling | Known attacks | High accuracy | Determine system risk |
| FCM | Identification of worst attack | Finite cognitive map | Genetic algorithm | Known attacks | Less complex | Less false positive rate |
| CBSH | Root cause of the attack identification | Design model | Probabilistic approach | Known attacks | Adaptability, scalability | Complexity O (MN2) |

**Table 5**
Summary of distributive network forensics techniques.

| Distributive | Objective | Forensic approach | Methodology | Detection of attack | Characteristics | Performance |
|---|---|---|---|---|---|---|
| ForNet | Distributive analysis | Bloom filter tracking | Logging | Known attacks | High response time, Light weight filtering | Trustworthy information |
| DCNFM | Integrity and validity of evidence | Mapping topology, network attack statistic | Logging | Known attacks | Classification, link and sequential analysis | Identify potential risk |
| DRNIFS | Real-time network intrusion analysis | Log and network traffic analysis | Logging | Known attacks | Robust, Less false positive rate | Quick incident response |
| DNF-IA | Integrity and authenticity for evidence | Multi-Immune theory | Logging | Known attacks | Scalable, high response | Real-time analysis of the attack |

**Table 6**
Summary of NFT using intrusion detection systems.

| Intrusion detection system | Objective | Forensic approach | Methodology | Detection of attack | Characteristics | Performance |
|---|---|---|---|---|---|---|
| AIDF | Identification of un-identify signature rule | Probabilistic model | Probabilistic inference | Unknown attacks | Scalability, extracting Hide information | Prefect discovery, flexible, robust |
| DFITM | Forensic server tolerance | Formal methods | Formal analysis | Known attacks | Separation of malicious traffic | Availability, high throughput, tolerance |
| IIFDH | Monitoring log files | Steganography | Logging | Known attacks | Integrity and correctness for evidence | Integrity of evidence with real-time detection |
| NFIDA | Credibility and reliability for evidence | Multi-dimensional analysis | Logging | Known attacks | Data encryption, multi-dimensional analysis | Capturing of complete network traffic |

it creates complexity in the overall system. It also lacks a management module for maintaining hash values. TDPM selects random hash fragments for packet marking which have more chance to miss appropriate fragments necessary to be marked for investigation process. In addition, TDPM does not capture application layer traffic that is more targeted nowadays by intruders such as voice over IP traffic.

Collaborative Forensics Scheme (CFS) is proposed to investigate VoIP traffic by using traceback technique (Hsu et al., 2011). The CFS investigates logs collected from Network Operators (NOs) and service providers rather than collected from routers. CFS saves time and storage by not performing packet marking and logging at the router level as performed in TDPM (Wang and Wang, 2010). CFS determines fake values embedded by intruder in the Session Initiation Protocol (SIP) request. Although, CFS determine forge header value on SIP based services and compares it with NOs and SIP registrar to traceback malicious traffic and provide further investigation. CFS traces application layer data i.e. VoIP data which is not investigated by TDPM framework (Wang and Wang, 2010). However, the critical aspect of CFS includes the non-identification of anonymous attacks that frequently occur in VoIP communication. Anonymous attacks help intruders to erase their footprint signs to help them in keeping their identity hidden. In addition, CFS reduces the performance due to it's completely dependency on NOs and SIP registrar data. CFS has no access towards data upon the denial of NOs and service providers to provide the collected

evidence due to privacy, organization policy, or some other competition aspect.

Network Forensic Evidence Acquisition (NFEA) scheme is proposed based on packet marking approach (Kim and Kim, 2011). It uses traceback to determine the origin of intruder with better effective tracking range and evidence. NFEA improves effective tracking range by looking towards router information and packet header while in Hsu et al. (2011) no router information was recorded which does not increase system throughput. The NFEA minimizes overall overhead of network throughput by tracking data at edge routers, whereas in Wang and Wang (2010) data is captured at all routers of the network. Data is collected from the edge routers because every packet which enters or leaves the network passes through it. The collected data is managed by trustful operators and it produces less overhead on edge routers and overall network performance. Edge router includes a time label for each record that helps in identifying step by step intruder activities. However, the critical aspect of NFEA is elaborated by finding origin of the attack when attacker hides its MAC address. Moreover, NFEA is also not adopted for IPv6 address due to its own packet structure. In addition, the router memory remains not sufficient when information table entries increase at the edge router. NFEA also lacks scalability and reliability in terms of large network systems.

Lightweight IP traceback scheme (LWIP) use Time-To-Live (TTL) field of IP header to investigate attacks in the network (Fen et al., 2012a). LWIP only records the packet information rather than packet marking as performed in Kim and Kim (2011) and Wang and Wang (2010). It targets large DDoS attacks wherein it finds and reconstructs the attack path. LWIP provides positive results even an intruder knows the victim is already running IP traceback scheme for its detection. The attack is traceback by performing tree analysis algorithm. In addition, it also use routing filter algorithm to overcomes the shortcoming of the storage problem (Kim and Kim, 2011) at router by filtering the data. The critical aspect of LWIP scheme includes the movement of each packet from router to router towards victim node by decrementing TTL value and passes it forward. TTL value can be altered by intruder and its value might reach to its minimum such as zero before it reaches to its destination. Furthermore, in huge data traffic networks high intellectual routers requires to perform such activity quickly and precisely to produce accurate and on time results for incident response (Inayat et al., 2015) which lacks in LWIP proposed framework. The LWIP framework only targets IPv4 headers whereas now a day in many networks IPv6 packets are used for the communication.

Scalable Network Forensic (Scalable-NF) scheme for stealthy self-propagating attacks is used to traceback origin of the intruder on the basis of long period of the historical network traffic traces (Chen et al., 2013a). Scalable-NF addresses two challenges that are identified in Wang et al. (2008) such as: (a) cyber-attacks increased and become complex, and (b) forensics analysts face with low data integrity at network traffics. The Scalable-NF is scalable in terms of space and time to provide high accuracy in determining stealthy self-attackers origin. A historical network traffic traces which is logged prior is divided into two parts including attack traffic and normal traffic. Attack traffic is generated by the intruder that is necessary for forensics network investigation, whereas normal traffic is the regular routine traffic that passes from different hosts to different end points without any interference of the intruder. Scalable-NF applies random moonwalk algorithm (Xie et al., 2005) on the attacked traffic which works has a forensic algorithm. The random moonwalk algorithm helps in traceback origin of the intruder by generating casual tree. In causal tree, the root node is considered to be the origin of intruder to whom forensic investigation is conducted. However, the critical aspects of Scalable-NF (Chen et al., 2013a) includes the biased result if it does not have significant information in the historical network traces. In addition, Scalable-NF incurs complexity due to utilization of extra resources to divide network traces in to normal and attack traffic. Furthermore, it also lacks an automation system to capture real time data and provide visualization as well to have a clear image of the entire scenario.

Hopping Based Spread Spectrum Technique (HB-SST) is proposed based upon code hopping-direct sequence spread spectrum, frequency hopping-direct sequence spread spectrum, and time hopping-direct sequence spread spectrum (Yu et al., 2013). HB-SST is used for network forensics traceback to tackle cybercrimes in anonymous communication of the network. However, a number of existing frameworks are proposed to help user in communicating anonymously in the network (Levine and Shields, 2002, Srinivasan, 2002). The HB-SST technique sends pseudo-noise code with a normal traffic which is invisible to the intruder. The malicious traffic of the intruder is monitored by the investigators through these secret pseudo-noise codes. However, scalability is considered a critical aspect of HB-SST in large data networks. The HB-SST complicates the investigation process by consuming more time in sending pseudo-noise codes within large data communication sets. Though, if these codes are identified by intruder, it can easily interfere and affect the communication by altering the codes. Furthermore, in frequency hopping-direct sequence spread spectrum users frequently shuffle within different frequency bands for optimizing their communication faster, it complicates the investigation process to monitor such type of transmission in a real-time situation.

IP traceback protocol (ITP) is proposed for network forensics analysis to determine origin of the intruder (Jeong and Lee, 2013). The ITP performs both real time and periodic analysis on the network traffic that passes from the router. Real time analysis is performed upon hash tables at the router, whereas periodic analysis is performed on compress hash tables. The compress hash tables generated at routers are store in a database which is used further by investigating network traffic. The ITP traceback attacks attributable to time-stamp and MAC field attached with the packet and keeps integrity by using hash function. Time-stamp value is used to retransmit the attack; whereas MAC field assist routers to know different devices through which packet moves through. It supports ITP to traceback attacks in real-time as well as periodically based on the situation. However, the critical aspect of ITP includes the system complexity for compressing and decompressing packets at router which results in more jitter. To investigate huge data in large distributed networks, many computational resources are required and it results in high time delays due to high latency and large devices present in the network. Moreover, ITP incurs problem of un-updated router attack pattern list that adversely affects the overall system. This affects system because new attacks are not traceable due to un-updated pattern list available at the router.

#### 3.2.2. Converge network based NFT

This section discusses the identification of digital evidence in converge networks especially in VoIP communication. VoIP communication inherits security threats, vulnerabilities, and attacks from data network as it uses a medium for its communication (Park, 2008). In VoIP communication signals are divided into frames which are embedded in data packets as voice codes and are sent on IP network as a normal voice packet. Voice packet sends from one caller (sender) to another caller (receiver) without any interference and modification by the intruder is called normal voice packet. Typically, SIP and H.3231 protocols are used to transport voice packet on the IP network (Johnston and Piscitello, 2006). Voice packet encloses IP addresses and ports information to

assists voice communication protocols that also act as a session control protocol. The enclosed IP addresses and ports information is not encrypted due to the requirement of the network address translation devices for translating the voice traffic. Thus, unencrypted fields in voice packets increase the probability to be attacked by intruders. Voice packet is exploited by intruders during voice communication which changes normal voice packet to the malicious packet (Park, 2008). Generally, malicious packets are generated by the intruder in different attacks including flooding, hijack calls, buffer overflow, man-in-the-middle, eavesdropping, privacy leakage, degrading call integrity, and exploitation of various VoIP devices.

Pattern Based Network Forensics (PBNF) technique is employed on VoIP network traffic (Pelaez and Fernandez, 2009). A pattern is known as generic systematic solution for the same type of problems occurring in different type of situations. In the previous solution (Fernandez et al., 2007), attack pattern is generated when attack occurred whereas PBNF employs forensic network pattern in VoIP networks to collect and analyze network traffic in a systematic way (Pelaez and Fernandez, 2009). It guides in finding and stopping attacks, determining traces of the intruder, and collect evidence about the attack to prosecute intruder in the court. PBNF is composed of two components including VoIP evidence collector and VoIP evidence analyzer. VoIP evidence collector is responsible to collect attack packets in real-time environment on the basis of filter rule set for the voice traffic. However, VoIP evidence analyzer analyzes collected data from forensic server through normalization (Forte, 2004) and log correlation; attack patterns are extracted from such data. Nevertheless, the scalability is considered to be critical aspects of PBNF due to finding exact number of sensors require collecting data in large distributed networks. The storage is another concern for PBNF that stores captured packets in both memory and forensic server. Forensic server is located centrally that becomes bottle neck for various attacks performed by intruders to exploit evidence store in it.

VoIP network forensics analysis with digital evidence procedure (VoIP-NFDE) identifies attack packets by differentiating normal and abnormal packets (Lin et al., 2010). Abnormal packets are also called affected or malicious packets which are injected in the network by intruders. A solution proposed in Lin et al. (2010) performs forensic investigation by collecting digital evidence. The digital evidence is collected by gathering information from packet protocol, service type, TTL value, and payload of the voice packet. Any alteration found in fields of the packet is considered to be malicious. The VoIP-NFDE (Lin et al., 2010) is different than (Pelaez and Fernandez, 2009) due to its focus on packet fields rather than packet logs. However, the critical aspect of VoIP-NFDE includes the scalability issue related to large distributed networks. VoIP-NFDE lacks to provide an appropriate solution to capture all disseminated voice packets in distributed networks. Moreover, the investigation process is more time consuming due to re-establishment of communication link between IP phone user and SIP registrar upon its disconnection. The re-establishment of communication link takes time and have chance to lose data which might be useful for investigating attacks. Similarly, huge storage resources required by storing attack patterns collected from voice data during investigating VoIP traffic.

VoIP Evidence Model (VoIPEM) investigates malicious attacks in VoIP communication by producing hypothesis basis on information gathered (Ibrahim et al., 2012). VoIPEM reconstructs attack scenario by using secure temporal logic of action which is also used by a method in Rekhis (2007). Secure temporal logic of action is used when insufficient details are available for investigating a system. It is generic and logic based formalized method used for any field that lacks significant information. This formalized method helps investigator to cover the unknown attacks that are not captured by the system. The VoIPEM works in four steps to investigate malicious attacks such as: (a) first, the evidence is provided to know about the crime scene, (b) second, the information is collected from all VoIP infrastructure components, (c) third, hypothesis is generated to identify more findings and evidences regarding the attack, (d) fourth, forward and backward chaining is used to identify evidence from attack states of the model. VoIPEM reconstructs whole attack scenario to clearly identify the location, procedure, and time of the attack. However, VoIPEM does not identify attacks in anonymous communication. The evidence is collected from all VoIP infrastructure components which require huge storage capacity to store massive amount of voice data. Furthermore, evidence generation is performed that totally depends on strong hypothesis, whereas hypothesis need to be intelligently generated, which requires enough information and awareness of the system.

#### 3.2.3. *Attack graphs based NFT*

Attack graphs are used to identify all possible attack paths in the network an intruder performed during its attack by analyzing hosts, networks, and other security devices (Ingols et al., 2006, Saha, 2008). Attack graph identifies and visualizes attack paths used by an intruder during their attacks. Attack graph contains vertices which represent attack nodes and edges identifying state transition between different vertices. Moreover, attack graphs are used for network forensics investigation which visualizes (Harbort et al., 2011) attack paths and determine worst attack paths (Diamah et al., 2012) situation to help network administrator in defending before an attack has been occurred. Attack graphs are used to identify scalable and impact analysis (Albanese et al., 2011), identify multi stage network attacks (Fen et al., 2012b), evidence collection (Liu et al., 2012), and cost-benefit security harden (Zhang et al., 2012).

The visualization and interaction framework is proposed for attack graphs to trace out intrusion behavior of attacks (Harbort et al., 2011). It is time consuming to determine relevant attack paths within thousands of vertices and edges in the attack graphs. Existing works (Liu et al., 2010; Ma et al., 2010; Ruia et al., 2011) depict attack graphs with different aspects including virtual exploitation information, attack dependency graph, data reduction, critical system, and others. However, little consideration is given to visualize attack graphs. To embed visualization features in the attack graph, RAVEN architecture (Harbort et al., 2011) is proposed to overcome the complexity of attack graphs in large networks and provide easy way for investigators to interact with its interfaces through visualize interfaces. RAVEN has wide array of gesture controls and provides collaborative analytical environment for multiple investigators. Furthermore, it has human-computer interaction platform that is designed to interact with attack graph in a user manageable way. Human-computer interaction multi-touch technology is combined with RAVEN to provide investigator natural experience while viewing each node and its transaction of the graph. However, the critical aspect of RAVEN includes non-real-time visual support in a real-time environment. It also lacks composite layout in its interface to assist users in looking towards multiple attack paths parallel and can reconstruct attack paths easily. The RAVEN also lacks deployment of data mining approaches for retrieving relevant information from the entire network efficiently.

Scalable Analysis (SA) approach determines larger amount of alerts to measure impact of the current and future malicious attacks in large network traffic environment (Albanese et al., 2011). Existing strategies have created automatic attack graph generation within network (Ebrahimi et al., 2011; Sanguankotchakorn and Dechasawatwong, 2011); however, they lack scalability analysis of attack graphs and measurement of evaluating its attack

patterns. The SA provides framework for evaluating huge amount of raw data in real-time situation. The framework works in three sequential steps such as: (a) first, an attack graph is generated to determine the dependency of network components and addresses the interaction between them, (b) second, attack graph definition (Li and Schmitz, 2009) is extended with time-span distribution to capture the probabilistic temporal attack behavior of the intruder, (c) third, attack scenario graphs are generated to bridge the gap between vulnerabilities and services affect by corresponding exploits to integrate dependencies and attack graphs. Nevertheless, the shortcomings of SA approach includes the lack of automatic generation of dependency graphs to reduce computational overhead and make system more efficient in minimizing time delays. Categorization in attack graph is another important issue which helps in providing quick attack path identification but it lacks in a SA approach. Furthermore, SA does not include the visualize interface (Harbort et al., 2011) to assists investigator in investigating attacks easily while visualizing different attack paths.

Attack Graph for Forensic Examination (AGFE) incorporates anti-forensics techniques with attack graphs to perform forensic investigation (Liu et al., 2012). Nevertheless, it is possible that attack graph miss some of the evidence in collecting data for identifying attack paths due to traces deleted by intruders or it is being altered after they are generated. The intruder delete its footprints from the network by using various anti-forensics techniques (Sullivan, 2011, Whitteker, 2008). It creates problem for forensic investigators in conducting their investigation to identify relevant traces of the attack. AGFE integrate anti-forensics nodes with attack graph to monitor intruder for its malicious attack. The intruder use anti-forensics nodes to reduce evidence; however it is monitored by the investigator to record malicious activities and store it inside anti-forensics database. Anti-forensics databases store anti-forensics activities in terms of attack type, cause and effect, tools, targeted nodes, access, and privileges of the intruder. This information helps investigator to easily analyze malicious activities of the attack and identifies alteration injected in traces. However, the critical aspect of AGFE includes the less scalability due to integration of anti-forensics nodes in large distributed networks. Anti-forensics nodes should be deployed in the sensitive part of the network to monitor intruder's malicious activities whereas it becomes difficult in DoS and DDoS attacks situation. Further, it also requires storage space to store and track each record of intruders when they are under observation. These constraints make system complex and time consuming for generating quick incident response.

Network attack modeling based on Multi-Level and Layer Attack Tree (MLL-AT) accurately models attack tree in multi stage network attacks (Fen et al., 2012b). MLL-AT determines system risk by evaluating system security threats which occurs due to different attacks. The MLL-AT is proposed due to deficiency found in existing attack tree approaches (Jin et al., 2010; Whitley et al., 2011). These deficiencies includes such as: (a) it creates confusion in determining attack traces and their result is not being compared, (b) single attack is addressed by having no correlation with other attacks, (c) attack nodes do not record any alterations that occurred due to their changed state, and (d) complexity in the network makes attack tree large. The MLL-AT uses attack tree where each node is assigned weightage and system risk is calculated by implementing multi-attribute utility theory (Kailiponi, 2010). This makes MLL-AT distinct as compared to the existing traditional attack trees approaches (Jin et al., 2010; Whitley et al., 2011). Once the system risk is calculated, unwanted nodes are removed and attack sequences are generated. Attack sequence is a combination of atomic attacks in the tree to identify the target of intruders. However, the shortcomings of MLL-AT includes by looking towards its scalability and storage capacity. In large distributed networks, thousands of nodes coordinate with each other to ensure communication and this makes attack tree more complex and time consumed. Moreover, weighting each node for different attack path in attack tree is difficult especially in the presence of DDoS attacks.

Fuzzy Cognitive Map (FCM) is developed from attack graph which employs genetic algorithm to identify attack paths (Diamah et al., 2012). In large networks, it is difficult to search attack paths that becomes complex when network grows and is difficult for network administrator to find its goal (Danforth, 2006). FCM helps network administrator to focus on paths that act as a worst case for network attacks rather than identifying all attack paths as performed in Albanese et al. (2011) and Fen et al. (2012b). FCM identifies worst attack path by performing three steps. First, attack graph nodes and edges are represented as concept nodes and edges in FCM proposed solution. Second, weights are assigned to different edges based on the influence of concept nodes in FCM that are stored in the adjacent matrix to calculate the effect of concepts and are used with FCM representation. Third, genetic algorithm based on values of adjacent matrix identifies the worst attack in the pool of attack behaviors. However, the critical aspect of FCM framework includes the interference of the human knowledge. To perform and develop fuzzy map one has to be fully familiarized with the knowledge and infrastructure of the network. Less knowledge and awareness about the current network infrastructure results in wrong direction that could be costly in terms of wrong selection of worst case of the attack. It is difficult to map up large network infrastructures, therefore FCM involves the issue of scalability and visualization interfaces to visualize all attack paths.

Probabilistic approach (Zhang et al., 2012) is proposed to identify cost-benefit security hardening (CBSH) by exploring attack graphs. It benefits network administrator to be aware of the root cause of an attack that can adversely affect the network. A number of approaches are proposed to find out the root cause of the attacks in networks such as attack tree (Ray and Poolsapassit, 2005; Fen et al., 2012b), and attack graphs (Sheyner et al., 2002; Liu et al., 2012); however adaptability, scalability, and accuracy are still addressable challenges. Hence, CBSH (Zhang et al., 2012) identifies root cause of vulnerabilities in a large network by providing cost effective security hardening for predefined cost and security metrics. This approach provides automation, adaptability, and scalability in terms of large traffic networks. CBSH works in three steps to identify cost-benefit security hardening. First, based on the observation the dependency graph is modified. Second, dependency graph is sent to Hidden Markov Model (HMM) for calculating probabilistic nature between observation and network states. Third, heuristic algorithm based on ant colony optimization (Cordon et al., 2002) is developed to identify root cause of the attack from the pool of countermeasure attacks. However, the shortcoming of CBSH includes the probabilistic approach which depends on human observation similarly like (Diamah et al., 2012) to assign probability values for different breaches identified in the network. High expertize is required regarding understandability of the network infrastructure to take a decision quickly and accurately. In addition, a tool such as MulWal (Ou et al., 2005) is used to generate attack graph is time consuming and non-adaptive especially for real-time network data traffic flows. Therefore, CSBH (Zhang et al., 2012) has to be adaptive for real-time scenarios to overcome the aforementioned constraints.

#### 3.2.4. Distributive based NFT

Distributive frameworks facilitate network investigators in overcoming the problem of scalability for NFT by distributing network forensics servers and data agent systems. Network forensics server collects data for analysis from distributed disperse

data agents at different locations in the network. Distributive NFT performs investigation (Shanmugasundaram et al., 2003), act on emerging responses (Anwar et al., 2015; Ren and Jin, 2005), identify origin of the attack (Ren, 2004), and perform evidence collection (Wang et al., 2007). However, distributive NFT creates an overhead to keep forensic server secured from intruders that are distributed in the network. Various distributive NFT are briefly discussed in subsequent paragraphs and are also illustrated in Table 5.

ForNet (Shanmugasundaram et al., 2003) is a distributed framework proposed to record network logs at different network devices and stores it at different places in a distributed network environment. Normally, packets and host logs are manually analyzed which create problems regarding low response time, improper logging mechanism, synchronization among logs, delay in response time. Nevertheless, ForNet solves the aforementioned problems and produces efficient evidence for forensic investigators. The SynApps is an application installed on each network devices such as switches and routers to collect the evidence from the network and it summarize the network events for a longer period of the time. Evidence of network vulnerabilities for different crimes is collected from the packet header that is feasible as compare with payload data (Ponec et al., 2010). ForNet searches IP connections, port connection records, and session creation among different hosts with the help of bloom filter tracking (Domingos and Hulten, 2000) to collect evidence against intruders attack. However, the critical aspect of ForNet framework includes the storage of raw network data for especially large networks. Similarly, ForNet uses lightweight intrusion detection system which increases more probability to undetect various attacks performed on the network. These attacks are in the form of DoS, sending rogue queries to forensic server for utilizing its resources, and packet modification by intruders. Furthermore, logs sent to the forensic server in insecure communication channels are at risk due to modification of log files performed by intruders.

Distributed Agent based Real Time Network Intrusion Forensics System (DRNIFS) (Ren and Jin, 2005) is adaptive technique used to produce quick response for attacks while investigating different logs and network traffic. DRNIFS overcomes the deficiency of non-real time frameworks (Shanmugasundaram et al., 2003; Ren, 2004) by working for real time scenarios. DRNIFS is composed of four modules including network agents, network monitors, network investigators, and network forensic servers. These modules coordinate to perform forensic investigation on a real-time basis to decreases data loss; whenever attacks occurred. Network agents are responsible for gathering information such as data extraction, digital signatures, and data encryption transport from host to network forensic server via secure sockets layer transport channel. Network monitor captures all network traffic and gives timestamp to each packet and stores it to the disk. Network forensic servers are used to change filter and dump rule, whereas network investigator is responsible to collect malicious events from different records. The working goal for network investigator is to determine origin, location, email address, name, physical address, cell number, and system configuration of the intruder. However, the shortcoming of DRNIFS technique includes the capturing of entire network traffic and log files from all distributed network agents. The collected data requires huge storage capacity and it might lose data integrity by storing at several locations. Moreover, forensic server remains as a bottle neck in the proposed framework that can be attacked by different intruders with various attacks. In addition, each four modules of DRNIFS framework communicate with network forensic server in an encrypted format of communication resulting in delays especially in real-time scenarios by decrypting the messages with consuming more resources. It will be better to have region wise network forensic servers to coordinate each other and provide load balancing in terms of forensic analysis by reducing overheads and complexities.

Distributed Cooperative Network Forensics Model (DCNFM) is proposed to identify potential risk, origin of intruder, and misbehavior of the packets (Ren, 2004). DCNFM model is based on client server architecture. Client agents are installed on different systems to capture network traffic in the form of logs from different distributive places such as firewalls, IDS, and system monitors. The log files are converted to a database file which is sent to the forensic server; whereas in Shanmugasundaram et al. (2003) log files were not converted to the database file. Forensic server performs sequential steps by mapping topology, filter and dumps traffic streams, converts traffic stream into database, mines forensics database, replies network behavior, performs network attack statistic, and produces visual report for the whole process. Forensic server uses different algorithms to do network forensics including link analysis, sequential analysis and classification. Link analysis is used to find correlation for the forensic data to discover various attack behaviors quickly. Sequential analysis is used to regenerate the attack to identify sequential steps of the attack to prevent them in the future. Classification is used to map the data to the predefined attack categories to assist network investigator in attack identification. However, the shortcoming of DCNFM includes the requirement of huge storage space to save massive amount of network data collected from distributed places of the network. The network logs are converted to the database file which might lose its integrity by changing the log format, which can be questionable in the court. Furthermore, forensic server acts as backbone in DCNFM and remains as a bottle neck to stands at risk for several attacks. In addition, capturing all network traffic makes DCNFM more time consuming and complex especially when operating in real-time situation.

Dynamical Network Forensics framework based on Immune Agent (DNF-IA) provides real time evidence for attacks in the network (Wang et al., 2007). Immune agent approach is used in network forensics for digital evidence authenticity, validity, and integrality. DNF-IA is divided into two main components such as forensic server and detector agent. Detector agent captures raw network traffic and performs antigenic presentation coding and keeps the message in its queue. The code is compared with non-self-code and upon match, detector agent identifies that intrusion behavior is being detected. A message is sent to the forensic server where it starts collecting digital evidence. Moreover, digital signature is developed with the help of hashing and it is further send towards forensic server. Forensic server performs analysis and creates replay for different attacks. However, the critical aspect of DNF-IA is noted due to its non-scalable approach for forensic server in large network infrastructure. It used single forensic server to analyze network traffic which restrict an approach to be adaptable for large networks with huge amount of the data. The forensic server becomes bottle neck in the investigation process which requires security to be safe from various intrusion attacks. However, the bottle neck problem of forensic server was solved in Nagesh (2006) by processing analysis on multiple forensic servers.

### 3.2.5. NFT using intrusion detection systems

The IDS protects and monitors networks from malicious attacks when an intruder tries to exploit the network (Jiang et al., 2000). The IDS triggers an alert message and inform management system about the detection of an attack. It used to detect malicious activities of an intruder that compromise the network integrity and confidentiality. Mostly, IDS used logging approach (Fan and Wang, 2010, Jiang et al., 2012) to describe forensic explanation (Sy, 2009), dynamic forensics (Chen et al., 2009), reliability of the evidence (Fan and Wang, 2010), and analyze network intrusion

data (Jiang et al., 2012). Various NFTs use to employ IDS as a method to assist investigation process by identifying breaches found in the network. Some of these techniques are briefly discussed in subsequent paragraphs below and illustrated in Table 6.

Analytical Intrusion Detection Framework (AIDF) proposed to integrates alert information from IDS sensors and forensic analysis integration for intrusion detection (Sy, 2009). This results in forensic explanation that is based on observed IDS alerts and unreported signature rule. AIDF narrows down intruder attacks by modeling their attacks and discloses hidden information through analysis by applying probabilistic approach. To detect attacks in the network traffic, AIDF compares network data packet with the pattern encoded in a signature rule. AIDF uses Snort (Roesch, 1999) for probabilistic inference process to trigger a rule upon a match of pattern encoded in a signature rule with network traffic. These rules help network investigator to detect and understand attacks and take necessary actions to prevent them in the future. However, the shortcoming of AIDF framework includes the lack of knowledge base modules to store untreated hidden data, which can act as a rule for future intrusion detection alerts. Untreated data checks for important information saves time in forensic investigation and produce precise results. This mechanism is used for real time situation to produce results with generating intrusion detection alerts.

Intrusion tolerance system is proposed for modeling and analyzing dynamic forensics system (DFITM) (Chen et al., 2009). Though a number of intrusion tolerance projects are proposed (Wang and Upppalli, 2003, Zhang et al., 2007), however there are less research efforts present related to intrusion tolerance. In most cases, forensic server is affected by attacks conducted by intruders that divert the system towards malicious states. Therefore, network or system is to be tolerable enough to work in such failure conditions. DFITM records real time alerts and works even when forensic server is affected by intruders (Sy, 2009) while performing forensic analysis. DFITM uses finite state machine to produce dynamic forensic system with intrusion tolerance. In addition, with finite state machine, Semi Markov process is developed base on embedded Markov chain to solve the aforementioned problems. Security attributes of the entire system are evaluated with the help of steady state probability of Markov model. However, the critical aspect of DFITM is viewed in terms of its storage capacity. Each data recorded is store in a multiple places that result in storage problem and also increase the risk for securing data at various places.

Intrusion investigation framework (IIFDH) provides data hiding for log files (Fan and Wang, 2010). The log files that are maintained at security devices are sent to the forensic server for investigating and collecting evidence from it. However, it is difficult to assure the integrity of log files that are sent to the forensic server for investigation (Chen et al., 2009; Sy, 2009). IIFDH employs a scheme based on steganography to hide original data logs from an intruder. Steganography is used to record log files that are altered by an intruder. In IIFDH (Fan and Wang, 2010), text is converted to the image and is kept as trace log and is hidden from the intruder. This is achieved by changing the least significant bit in each pixel that is not visible to intruders (Yang et al., 2008). This is a simple way to keep the log files hidden from intruders. However, IIFDH lacks to solves the bottle neck problem of forensic server, that is addressed through fault tolerance mechanism in Chen et al. (2009). Forensic server resides at centralized location, and it can be affected by malicious attack of intruders which further propagates its malicious affect throughout the network. In addition, forensic server bears processing and storage problem for log files that are sent by distributed devices from different locations in the network. IIFDH requires large bandwidth to communicate between systems and forensic servers for quick response in the real time investigation. Moreover, IIFDH requires incorporating security measures for protecting important components of the network from intruder attacks.

Network Forensics based on Intrusion Detection Analysis (NFIDA) implements static and dynamic analysis for network intrusion data (Jiang et al., 2012). NFIDA framework records network logs and provides credibility and reliability in providing network evidences through network forensics technique. A multi-dimensional forensic analysis is performed on the captured data of the network, which provides evidence against intruders. Behavior of the intruder is captured to analyze during digital forensics investigation process. The NFIDA performs network analysis in offline mode by using pattern matching and protocol analysis. Pattern matching technique analyzes packets based on library traces; whereas protocol analysis technique analyzes type of protocol in the packet and implements data analysis procedures to identify malicious behavior of packets. Nevertheless, the critical aspect of NFIDA framework includes the lack of real-time network analysis for the network traffic. The NFIDA incorporates scalability issue which restricts NFIDA to be fully adopted for large distributed networks. Moreover, the newly collected network traffic in storage space overwrites existing data in memory that result in evidence lose with effecting entire investigation process.

## 4. Comparison of network forensics techniques

This section presents a comparison of current NFT based on the parameters presented in Fig. 1. The comparison parameters include mechanism, target dataset, target instance, forensic processing, time of investigation, execution definition, and objective function. A comprehensive discussion on the comparison parameters is presented as follow.

### 4.1. Mechanism

The NFT use various mechanisms to perform network investigation for identifying vulnerabilities in the network. The investigation process of various mechanisms is based on the information of network logs, network packets, and various network events of the network. Such information provides valuable insight into various mechanisms and conduct investigation by providing evidence from the network. For such reason, various mechanism used by NFT is categorized under the heading of logging, packet marking, and heuristic base respectively. Each of the mechanism is mentioned below in the subsequent paragraphs and is briefly explained with its respective NFT in Section 3.2.

***Logging (LO):*** Logging is an approach used to record network flows and patterns in database to determine evidences regarding attacks. It works by storing source IP address, destination address, number of packets, and various others depending on the situation. This information is captured at variant routers and is used in tracing out attacks (Gong and Sarac, 2005; Takemori et al., 2008). In addition, various mechanisms use network logs to do network forensics including random moonwalk algorithm (Xie et al., 2005), Apriori algorithm (Liu, ,2010), hypothesis generation (Ibrahim et al., 2012), immune approach (Wang et al., 2007), steganography (Fan and Wang, 2010), and pattern and protocol analysis (Jiang et al., 2012). Random moonwalk algorithm identifies frequent causal edges in the contact graph generated from network records; construct a causal tree which represents an attack node at its root node. Apriori algorithm is used to determine associated rules in attack pattern based on existing logs collected at various stages. These rules help in identifying more attacks in the future. Hypothesis generation is performed based on the information gathered in the form of logs to identify evidence against attacks in

the network. Immune agent provides authenticity, validity, and integrality for the digital evidence collected in the form of logs from the network. Steganography is used to hide log traces from intruders and monitor their activities seamlessly. Pattern analysis is used to analyze existing traces to determine pattern matching whereas protocol analysis is used to analyze protocol type in the packet. Both of these analysis helps in identifying the malicious behavior of the packet in the network. Moreover, hashing is performed to log network packets at different network devices as mention by NIST i.e., routers, firewalls, host IDS, and network IDS (Kent and Souppaya, 2006). These devices identify security incidence, attack patterns, operational problem, and different drifts in the network traffic. However, logging faces various challenges in terms of less storage capacity to store all network flows, protecting security devices, and fast computation at the point of huge network traffic flows. Hence, as shown in Table 7, NFT that employs logging mechanism is represented with the value "Yes", whereas "No" represents the employment of other mechanisms besides logging.

***Packet Marking (PM)*:** The PM is an approach to mark network packets at different routers during network flow from sender to its destination (Akyuz and Sogukpinar, 2009; Gong and Sarac, 2005; Siris and Stavrakis, 2007; Yonghui et al., 2010), and is often used by IP traceback techniques to identify sender IP address that is spoofed by an intruders (Kim and Kim, 2011). The PM becomes problematic when intruder sends huge amount of packets, aware of tracing mechanism used, and routers low memory. In addition, various mechanisms use marked packets to traceback attacks i.e. Authenticated Evidence Marking Scheme (AEMS) (Kim and Kim, 2011), tree analysis algorithm (Fen et al., 2012a), and Probabilistic Packet Marking (PPM) (Bo et al., 2009). AEMS is used to append encoding code with the packet at the router level so that it can be trace later in the network. Tree analysis algorithm created from marked packet identifies attack paths in the tree by comparing the values of child node at the same level, values that is assigned in the previous stages of the algorithm. The PPM is used to mark packet at the router based on start, end, and distance parameters. Start is the point where the packet is initiated, end is the destination point of the packet, and distance is the number of hops present between start and end point of the packet. However, there are three types of packet marking techniques, they are: (a) deterministic packet marking is used to mark a packet at the ingress router in the network (Dou et al., 2012; Qu and Huang, 2008; Wang and Wang, 2010; Yonghui et al., 2010). It is simple and less complex as compared to other packet marking mechanisms (Fen et al., 2012a; Kim and Kim, 2011). It further determines sender IP address by receiving few marked packet at the victim nodes; however, it is poor in identifying DDoS attacks (Wang and Wang, 2010). (b) The PPM is used to mark packets at every router during its travel from sender to the destination (Bo et al., 2009; Fen et al., 2012a; Kim and Kim, 2011; Kok and Kurz, 2011; Ren and Jin, 2005; Xiao-jing and Sheng-jun, 2009). It injects additional bits in IP address field and increments hop counter at each router from where the packet moves on (Emre, 2011). This process is complex and time consuming; however, it traces out attack paths more accurately. (c) Internet control message protocol traceback also called iTrace is used to send Internet Control Message Protocol (ICMP) packet with its header, router and neighbor ID on the normal path where packets move on (Jeong and Lee, 2013). It gathers information regarding adjacent routers and embeds it in its message. Such messages are further used by the victim node to reconstruct attack paths. In Table 7, a NFT which deploys packet marking mechanism is shown with the value of "Yes"; whereas, "No" represents the employment of other techniques rather than packet marking. The following NFTs (Fen et al., 2012a, Kim and Kim, 2011, Ren and Jin, 2005) use packet marking mechanism to perform network forensics investigation.

***Heuristic Based (HB)*:** Heuristic based mechanisms are used to observe and solve the problems based on the network information. For the sake of ease, various schemes i.e., spread spectrum technique (Yu et al., 2013), immune theory (Wang et al., 2007), attack graphs (Albanese et al., 2011; Liu et al., 2012), finite state machine (Chen et al., 2009), Hidden Markov model (Zhang et al., 2012), and fuzzy cognitive maps (FCM) (Diamah et al., 2012) are integrated under heuristic based title for our analysis. Spread spectrum technique generates disperse signal with certain bandwidth at a frequency in the domain for different reasons including security, detection, traceback, and finding receiver status. In Yu et al. (2013), three types of spread spectrum techniques are discussed which include code hopping-direct sequence spread spectrum, frequency hopping-direct sequence spread spectrum, and time hopping-direct sequence spread spectrum. In addition, finite state machine (Chen et al., 2009) is used to address states of the machine in the network, whereas performing tolerance of the intrusion systems. A system can be in one state at a time and changes its position from one state to another and it also addresses physical resources in terms of the logical representation. Furthermore, it results in providing forensic server availability and capability to cope with the failure situation (Khan et al., 2012). Moreover, Hidden Markov Model is a statistical model used to address hidden states with a visible output depends on its transition. In Zhang et al. (2012), Hidden Markov Model is used to determine probabilistic relation between states and observations which helps in identifying various attack states that were not being discuss in Yu et al. (2013). Furthermore, fuzzy cognitive maps are used to deal with the causal reasoning (Yonghui et al., 2010) among nodes of the network, whereas in Diamah et al. (2012) it only identifies worst attack states for the attack. The NFT employing heuristic base mechanism are represented with the value of "Yes", whereas "No" represents the deployment of techniques other than heuristic base.

### 4.2. Target dataset

Target Dataset (TD) attribute shows the type of the data which is targeted by NFT. For instance, data traffic (Jeong and Lee, 2013; Boruah and Hazarika, 2014; Cheng et al., 2013; Fan and Wang, 2010; Jiang et al., 2012; Wang et al., 2007; Zhang and Wang, 2013) converge network such as VoIP data (Ibrahim et al., 2012; Lin et al., 2010; Pelaez and Fernandez, 2009), and network model (Albanese et al., 2011; Fen et al., 2012b; Jemili et al., 2007; Kruegel et al., 2003; Liu et al., 2012; Zhang et al., 2012). A number of current NFT (Boruah and Hazarika, 2014; Chen et al., 2009; Chen et al., 2013a; Cheng et al., 2013; Fan and Wang, 2010; Fen et al., 2012a; Jeong and Lee, 2013; Jiang et al., 2012; Kim and Kim, 2011; Ren, 2004; Ren and Jin, 2005; Shanmugasundaram et al., 2003; Sy, 2009; Wang et al., 2007; Yu et al., 2013; Zhang and Wang, 2013) target data traffic that contains IP packets and they are treated by using logging (Chen et al., 2013a; Fan and Wang, 2010; Ibrahim et al., 2012; Jeong and Lee, 2013; Jiang et al., 2012; Kim and Kim, 2011; Lin et al., 2010; Pelaez and Fernandez, 2009; Ren, 2004; Ren and Jin, 2005; Shanmugasundaram et al., 2003; Wang et al., 2007), packet marking (Cheng et al., 2013; Fen et al., 2012a; Kim and Kim, 2011; Ren and Jin, 2005) or heuristics based (Albanese et al., 2011; Boruah and Hazarika, 2014; Chen et al., 2009; Chen et al., 2013a; Cheng et al., 2013; Diamah et al., 2012; Fen et al., 2012b; Harbort et al., 2011; Ibrahim et al., 2012; Jemili et al., 2007; Kruegel et al., 2003; Liu et al., 2012; Sy, 2009; Wang et al., 2007; Yu et al., 2013; Zhang and Wang, 2013; Zhang et al., 2012) mechanism. Nevertheless, voice packets are traced and captured when voice packet travels across the Internet i.e. in the case of IP telephonic calls and

**Table 7**
Analysis of network forensics techniques based on thematic taxonomy parameters.

| Frameworks | | Mechanisms | | | TD | TI | FP | ToI | ED | OB |
|---|---|---|---|---|---|---|---|---|---|---|
| | | LO | PM | HB | | | | | | |
| **Trace back** | NFEA (Kim and Kim, 2011) | No | Yes | No | Data traffic | Packet header | Centralized | Static | Proactive | Origin of attack |
| | LWIP (Fen et al., 2012a) | No | Yes | No | Data traffic | Packet header | Centralized | Dynamic | Proactive | Origin of attack |
| | Scalable NF (Chen et al., 2013a) | Yes | No | Yes | Data traffic | Complete packet | Centralized | Dynamic | Proactive | Origin of attack |
| | HB-SST (Yu et al., 2013) | No | No | Yes | Data traffic | Complete packet | Decentralized | Dynamic | Proactive | Origin of attack |
| | ITP (Jeong and Lee, 2013) | Yes | No | No | Data traffic | Packet header | Centralized | Static | Reactive | Origin of attack |
| **Converge network** | PBNF (Pelaez and Fernandez, 2009) | Yes | No | No | VoIP | Packet header | Centralized | Dynamic | Reactive | Real-time analysis |
| | VoIP-NFDE (Lin et al., 2010) | Yes | No | No | VoIP | Packet header | Centralized | Dynamic | Reactive | Filtering network traffic |
| | VoIPEM (Ibrahim et al., 2012) | Yes | No | Yes | VoIP | Complete packet | Centralized | Static | Reactive | Reconstruction of attacks |
| **Attack graph** | SA (Albanese et al., 2011) | No | No | Yes | Network model | Network nodes | Centralized | Static | Reactive | Scalable and impact analysis |
| | MLL-AT (Fen et al., 2012b) | No | No | Yes | Network model | Network nodes | Centralized | Dynamic | Reactive | Identify multi-stage n/w attack |
| | AGFE (Liu et al., 2012) | No | No | Yes | Network model | Network nodes | Centralized | Dynamic | Reactive | Evidence collection |
| | FCM (Diamah et al., 2012) | No | No | Yes | Network model | Network nodes | Centralized | Static | Reactive | Identify worst attack |
| | CSBH (Zhang et al., 2012) | No | No | Yes | Network model | Network nodes | Centralized | Static | Reactive | Cost-benefit security harden |
| | AGVI (Harbort et al., 2011) | No | No | Yes | Network model | Network nodes | Centralized | Static | Reactive | Visualization |
| **Distribution** | ForNET (Shanmugasundaram et al., 2003) | Yes | No | No | Data traffic | Packet header | Decentralized | Static | Proactive | Investigation |
| | DRNIFS (Ren and Jin, 2005) | Yes | Yes | No | Data traffic | Complete packet | Decentralized | Dynamic | Reactive | Emergence response |
| | DCNFM (Ren, 2004) | Yes | No | No | Data traffic | Complete packet | Decentralized | Static | Proactive | Origin of attack |
| | DNF-IA (Wang et al., 2007) | Yes | No | Yes | Data traffic | Packet header | Decentralized | Dynamic | Reactive | Evidence collection |
| **Intrusion detection system** | AIDF (Sy, 2009) | No | No | Yes | Data traffic | Complete packet | Decentralized | Static | Reactive | Forensic explanation |
| | DFITM (Chen et al., 2009) | No | No | Yes | Data traffic | Complete packet | Centralized | Dynamic | Reactive | Dynamic forensics |
| | IIFDH (Fan and Wang, 2010) | Yes | No | No | Data traffic | Complete packet | Centralized | Dynamic | Proactive | Reliability of evidence |
| | NFIDA (Jiang et al., 2012) | Yes | No | No | Data traffic | Packet header | Centralized | Static | Reactive | Analyze network intrusion data |

**LO**: Logging; **PM**: Packet marking; **HB**: Heuristic based; **ED**: Execution definition; **TD**: Target datasets;
**TI**: Target instance; **FP**: Forensic processing; **ToI**: Time of investigation; **OB**: Objective function

several voice messengers including Skype. In addition, attack graphs use network models which identify attack paths by determining various attacked network nodes. Network models contain network topology and configuration data to each network node present in the network. Such network model helps in developing attack graphs by integrating various attack rules with it.

### 4.3. Target instance

Target Instance (TI) represents the type of instance which is targeted by NFT to identify digital evidence. These instances include complete packet, packet header, and network nodes. A packet contains two portions including payload portion and packet header portion. Packet payload contains the user data which is sent from the source to the destination, which can be changed by intruders on the way. Packet header contains the routing information for sending the packet from the sender towards its destination. Complete packet is captured by NFT for looking at the entire packet to identify malicious activity of an intruder. Moreover, in addressing all the fields of the packet, it results in longer processing delays but it precisely identifies intruders attack with solid evidence. Complete packet targeted by NFT (Chen et al., 2013a; Fan and Wang, 2010; Ibrahim et al., 2012; Ren and Jin, 2005) is used to identify pattern matching and further it is used to reconstruct several attack paths.

However, in investigating packet header, it does not check the payload part of the packet while just focusing on the header of the packet. A number of NFT targets packet header (Boruah and Hazarika, 2014; Cheng et al., 2013, Fen et al., 2012a; Jiang et al., 2012; Kim and Kim, 2011; Pelaez and Fernandez, 2009; Shanmugasundaram et al., 2003; Wang et al., 2007; Zhang and Wang, 2013) in investigating network traffic through IP source address, destination address, mac address and protocol type (Goralski, 2009). The packet header contains such information to help forensic investigators in identifying intruders by tracing the packet paths.

In addition, network nodes used by attack graphs represents network devices in the network that act as apart in the attack. Attack graphs treat such network nodes by finding interdependency between them, leading them to identifying various attack paths. Attack paths are useful to determine origin of the intruder and its footprints to further illustrate damages performed in the network. Several NFT (Albanese et al., 2011; Diamah et al., 2012; Fen et al., 2012b; Harbort et al., 2011; Jemili et al., 2007; Kruegel et al., 2003; Liu et al., 2012; Zhang et al., 2012) were studied and analyzed to used network nodes as a target instance for performing network forensics in our literature.

### 4.4. Forensic processing

Forensic Processing (FP) attribute depicts the way network forensics takes places according to its location such as centralized or decentralized. A number of current NFT used centralized processing (Fan and Wang, 2010; Fen et al., 2012a; Harbort et al., 2011; Jiang et al., 2012; Kruegel et al., 2003; Lin et al., 2010; Liu et al., 2012; Zhang and Wang, 2013). In centralized forensic, a single forensic server is accessed by different network nodes or agents locally or remotely for storing, querying, analyzing, and presenting the evidence regarding intruder behavior capture from various part of the network. Advantages of centralized FP include quick response time, less time delays, and having central focus. Moreover, the shortcomings of centralized FP include lack of scalability, more focus by attackers, and single point of failure.

However, decentralized FP overcomes the limitations of centralized forensic by distributing forensic server in the network. Distributive NFT (Ren, 2004; Ren and Jin, 2005; Shanmugasundaram et al., 2003; Sy, 2009; Wang et al., 2007; Yu et al., 2013) employ decentralized FP, however it has several limitations that includes less consistent, time delays, lack of centralized control, and synchronizing among distributed data. In addition, it is more complexity and has higher overhead especially in large distributed networks. It delays the investigation process by capturing, recording, and analyzing network data from several locations; which further utilize more resources. High bandwidth communication channels are required to send captured logs from various places to the forensic server for investigating it with a quick incidence response especially in real-time situations.

### 4.5. Time of investigation

Time of Investigation (ToI) attribute illustrates either network forensics is performed statically or dynamically. Static forensics (Albanese et al., 2011; Cheng et al., 2013; Diamah et al., 2012; Harbort et al., 2011; Ibrahim et al., 2012; Jeong and Lee, 2013; Jiang et al., 2012; Kim and Kim, 2011; Ren, 2004; Shanmugasundaram et al., 2003; Sy, 2009; Zhang and Wang, 2013; Zhang et al., 2012) deals with the investigation that is performed after the attack. It has several advantages, i.e. tracing out each and every event properly from the network logs and trace out intruders activities briefly and accurately. Currently, network data travels with high data rate and millions of packets per second are generated overwrites existing data due to the lack of the storage capacity. In addition, there is no guarantee that either the data is or not altered by the intruder when the network forensics investigator collects network logs after the attack.

However, dynamic forensics is also called live forensics that capture, record, and analyze network data at the time of its flow (Boruah and Hazarika, 2014; Chen et al., 2009; Chen et al., 2013a; Fan and Wang, 2010; Fen et al., 2012a; Fen et al., 2012b; Jemili et al., 2007; Kruegel et al., 2003; Lin et al., 2010; Liu et al., 2012; Pelaez and Fernandez, 2009; Ren and Jin, 2005; Wang et al., 2007; Yu et al., 2013). It is useful for large distributed networks for a reason of volatility of the network traffic. Moreover, it overcomes static forensic problem that includes not analyzing data at the time of network flow. However, dynamic forensics requires computational resources for collecting, recording, and analyzing data at its run time. Similarly, it requires a large amount of data storage space to store marked packets while analyzing equivalently new injected packets coming into the network.

### 4.6. Execution definition

Execution Definition (ED) refers to the type of approach used for investigation in NFT. These approaches are divided into two main categories, i.e. proactive and reactive. Both of these categories are explained below.

***Proactive*:** Proactive approach is used by NFT to investigate the incident in real-time by providing automation to the system while minimizing user intervention (Alharbi et al., 2011). Automation incorporates three basic tasks: (1) collection of evidence, (2) generating relative triggers for an event, and (3) preservation and analysis of the digital evidence to identify attacks. At the end of proactive investigation, preliminarily report is generated to describe how the evidence was collected and analyzed against specific attacks or incident (Orebaugh, 2006). This report helps the investigator to defend the evidence against the attacks in the court. The attacks are analyzed by using forensic techniques using machine learning, data mining algorithms to support and build initial hypothesis of the incident (Alharbi et al., 2011). Proactive NFT provides more reliable and accurate evidence in real-time when the attack is happening (Garfinkel, 2010). It provides early

detection of network attacks and reduces the chance of deleting evidence by intruders after the attack. However, proactive NFT increases processing and storage overheads in terms of identifying attack patterns and preserving the evidence in real-time (Mitropoulos et al., 2005).

Additionally, NFT perform real-time analysis (dynamic) which act proactively by analyzing attacks on its occurrence or collecting the evidence against attacks from the network without compromising its integrity (Boruah and Hazarika, 2014; Chen et al., 2009; Fan and Wang, 2010; Fen et al., 2012a; Fen et al., 2012b; Jemili et al., 2007; Jeong and Lee, 2013; Kruegel et al., 2003; Lin et al., 2010; Liu et al., 2012; Pelaez and Fernandez, 2009; Ren and Jin, 2005; Wang et al., 2007; Yu et al., 2013). Traceback proactive NFT collect the information about the malicious packets soon after they have detected it and to start analysis in real-time to trace origin of the attack (Chen et al., 2013a; Fen et al., 2012a; Yu et al., 2013). Whereas in Liu et al. (2012), anti-forensics technique is used with attack graph to proactively monitor intruder for its malicious attacks. Meanwhile the intruders are oblivious to such monitoring by the investigator, helping the investigator to identify the source of the attack (Khan et al., 2015). Moreover, distributive NFT are used to capture real-time network traffic from disperse security devices and perform analysis to minimize the malicious effect of the attack on the network (Ren and Jin, 2005; Wang et al., 2007). Similarly, IDS also generates alerts and performs real time analysis by integrating forensic servers in the network (Chen et al., 2009; Fan and Wang, 2010). It trace out attack scenarios quickly in real-time to determine the type and nature of the attack.

***Reactive*:** Reactive approach of NFT is a post mortem approach to investigate an attack after it has occurred (Palmer, 2001). Reactive NFT investigates network vulnerabilities by identifying, preserving, collecting, and analyzing digital evidence extracted from the network. The goal of reactive NFT is to determine root cause of the attack (Khan et al., 2014a), correlate intruder to the attack, minimize the effect of the attack, and successfully investigate the malicious incident (Grobler et al., 2010). Reactive NFT reduce processing overhead by performing offline analysis of logs generated at different security devices. However, it is more time consuming because it analyzes complete log files generated during network flows at various network security devices including firewalls and IDS. Moreover, attackers use anti-forensics techniques to delete traces after performing an attack to sophisticates the situation for reactive NFT in collecting evidence accurately (Whitteker, 2008).

Additionally, reactive NFT traceback attacks by analyzing different logs generated at various security devices (Jeong and Lee, 2013; Kim and Kim, 2011). Logs are used to regenerate the attack in a sequential manner as if it was originally performed by the attacker. Similarly, attack graphs computes attack path by addressing various network states used by the attacker (Albanese et al., 2011; Diamah et al., 2012; Harbort et al., 2011; Zhang et al., 2012). It assists investigator in determining potential paths used by the attacker. These attack paths are eliminated in future to make it difficult for the attacker to attack again. Likewise, distributive NFT reacts when an alert is generated by a remote security device (firewall, IDS, etc.). Later, network traffic is captured from these devices in terms of logs and is analyzed while maintaining its integrity (Ren, 2004; Shanmugasundaram et al., 2003). Moreover, IDS also act has a means of reactive network forensics by generating alerts and sending logs to the forensic servers (Jiang et al., 2012; Sy, 2009).

### 4.7. Objective function

Objective function (OB) attribute shows the purpose of proposing a NFT. The ultimate goal of each NFT is to investigate malicious activity of intruders in the network. Different objectives of the network forensics includes origin of attack (Fen et al., 2012a; Jeong and Lee, 2013; Kim and Kim, 2011; Ren, 2004; Yu et al., 2013), visualizing the attacks (Harbort et al., 2011), reconstructing the attacks (Ibrahim et al., 2012), forensic explanation (Sy, 2009), dynamic forensics (Chen et al., 2009), reliability of evidence (Fan and Wang, 2010), analyzing intrusion data (Jiang et al., 2012), scalable and impact analysis (Albanese et al., 2011), identifying multi-stage network attacks (Fen et al., 2012b), evidence collection (Liu et al., 2012; Wang et al., 2007), identifying worst attacks (Diamah et al., 2012), cost-benefit security harden (Zhang et al., 2012), event classification (Kruegel et al., 2003), evidence reduction (Cheng et al., 2013), signature recognition (Jemili et al., 2007), prevention of novel attacks (Boruah and Hazarika, 2014), and effective feature selection (Zhang and Wang, 2013).

Traceback NFT (Chen et al., 2013a; Fen et al., 2012a; Jeong and Lee, 2013; Kim and Kim, 2011; Yu et al., 2013) are highly concerned about identifying origin of the attack by collecting intruder traces from the network. In converged network, NFT main objective function is to identify intruders by investigating different pattern match found in the network (Ibrahim et al., 2012, Lin et al., 2010). However, attack graphs are used to identify attack paths in the network that had been used by an intruder during its attack (Zhang et al., 2012). Distributive forensics assists in collecting evidence from distributive devices in the network which helps in investigating criminals for their suspicious activities (Wang et al., 2007). The IDS are mostly used to work as a catalyst in the forensic process by detecting malicious attacks in the network and log them; which can be further investigated (Jiang et al., 2012).

Table 7 shows that network forensics can be achieved in different ways and it can achieve different objectives and help network forensics investigators during their investigation. Traceback techniques (Chen et al., 2013a; Fen et al., 2012a; Jeong and Lee, 2013; Kim and Kim, 2011; Yu et al., 2013) are mostly proactive in nature and are used to identify origin of the attack from where the malicious packets are originated. It can easily identify criminals based on their suspicious acts by tracing out their origin. Moreover, in many situations it uses packet header as a target instance to perform the investigation (Fen et al., 2012a; Jeong and Lee, 2013; Kim and Kim, 2011). Large network contains more devices that delay traceback due to focusing on a large number of network devices. However, converged networks are reactive in nature and mostly voice packets are captured and recorded from several servers located in their infrastructure by using logging mechanism (Ibrahim et al., 2012; Lin et al., 2010; Pelaez and Fernandez, 2009). Logs are investigated to identify different pattern match for malicious behavior dynamically. Converged networks are now more targeted by intruders due to insecure voice communication channels. Voice traffic is attacked by intruders to leak out user confidentiality and privacy. Converged network requires complete strategy that should make VoIP communication secure and more robust to safe user communication from several attacks.

In addition, attack graphs provide modeling for attack paths in the network by addressing network nodes. It uses heuristic base approach to identify different attack paths that has been used by intruders for their attacks. Attack graphs are performed centrally with targeting distributive network nodes (Diamah et al., 2012; Liu et al., 2012; Zhang et al., 2012). Attack graphs requires an integrated visualizing functionality to help investigators in analyzing the attack path more easily (Harbort et al., 2011). Distributive network forensics overcomes the problem of single bottle neck found in many NFT by using multiple distributed forensic servers. It uses all the mechanism, i.e., logging (Ren, 2004), packet marking (Ren and Jin, 2005) and heuristic base approach (Wang et al., 2007) to investigate network traffic in distributed network environment. It employs live forensics (proactive) as well as static

forensics (reactive) depending on the situational demands. Newly emerged network infrastructure places greater demands on distributive forensics due to the disseminated nature of the networks. Distributive network forensics is useful in providing a wider scope for investigators in terms of capturing digital evidence from all part of the network.

The IDS is used for multiple purposes including detection, alert generation, analysis, and preventing different attacks in the network. Its multiple purpose functionality allows IDS to capture complete packet of the network traffic proactively and reactively. Usually, IDS devices are located in a place where all the network traffic passes through it, which are logged rather than packet marked. The packet marking is difficult when IDS is located where all network traffic passes to it. IDS are useful for investigating huge amount of network traffic by using logging mechanism.

## 5. Open challenges in network forensics

This section discusses network forensics challenges that are significant in investigating different types of distributed networks such as software defined networks (Khan et al., 2016a) and cloud computing (Gani et al., 2014). Fig. 2 depicts open challenges and Table 8 highlights the possible solutions for the particular challenge.

### 5.1. High speed data transmission

High data rate of network traffic creates difficulties for network forensics in capturing and preserving all network packets (Clegg et al., 2013). Millions of packets are transmitted over the network in no time, which passes through thousands of interconnected network devices. Such network devices act as evidence for network forensics to investigate susceptibilities by analyzing network data flow through it. Network susceptibilities in high speed data network require recording of all packets without losing any of them, which is a challenging and time consuming task. Most companies expand and enhance their network structure by connecting several distributive infrastructures with high speed networks, i.e. 10 GB of data or more (Clegg et al., 2013). Thus, most of the time network traffic is not completely captured by security devices, which results in incomplete logs for network data flows. These incomplete logs make reconstruction of suspicious attack more difficult and therefore identification of the origin of intruder becomes difficult. To overcome network forensics problem related to high speed packets transmission, an alternate solution that can capture, preserve, index and analyze such packets in real time situations is required.

To overcome the aforementioned problems, three different solution are proposed including hardware based (Tripathi, 2009), software based (Deri, 2005) and distributive based solution (Morariu and Stiller, 2008). In hardware based solutions, a specialized data capturing device is deployed to capture high speed network traffic by analyzing its packet in FPGA programmable processing element. It helps in filtering of specific data and performs real time analysis with quick incident response. However, nCap library can be used as software based solution to capture high speed network traffic by programming customized hardware. It is deployed in the user space rather than using the kernel space. It assists in active and passive network monitoring and can be easily deployed in user space by programmers. In addition, distributed packet capturing techniques are used to capture high speed network traffic to provide load balancing within different disseminated nodes and reduce the cost of memory and CPU cycles.

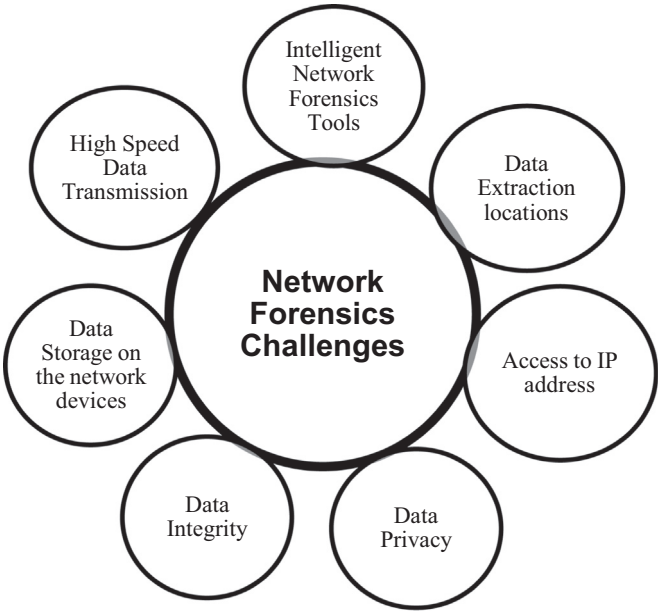


**Fig. 2.** Network forensics challenges.

### 5.2. Data storage on the network devices

A huge amount of data is transmitted over the network which is captured and analyzed for investigation. However, such data complicates the situation for network forensics to retrieve evidence from the network (Wang and Daniels, 2008). For instance, the captured data needs to be stored on devices with large storage capacity; whereas the storage capacity of the network interconnectivity devices is limited. A framework is proposed based on time machine packet capturing which reduces the problem of archiving the sheer amount of data on the network (Chen et al., 2013b). It performs fast bit indexing and querying the system based on flow granularity. It minimizes the storage requirements and speeds up the querying operation for high speed network data flows. Moreover, indexing packet is offloaded through general processing units (GPU) with performing compressed bitmap index in a real-time (Fusco et al., 2013). Deterministic operation of GPU helps in highly parallel operations by having faster and advance memory interfaces. It can easily store up to 185 million records per second. Additionally, n2disk architecture is also proposed to capture packet to disk application at the line rate on commodity hardware (Deri et al., 2013). It is used for single and multiple thread packet consumers in real-time situation and search specific packet in the network dump files. It reduces the storage problem especially in high speed network traffic i.e., for 10 gigabits per seconds.

Nevertheless, storage problems are generated due to the location of data captured in the network that is reported to capture network packets from several routers, IDS, firewalls, servers, hosts and others. Each of these devices must contain enough storage space to store network packets which enters the network with high data speed rate. In most of the cases, devices incorporate small buffer size and cannot accommodate such huge amount of network packets especially in heavily loaded networks. Thus, storage space for capturing network packets at various locations of the network is an important challenge which has to be workout to beneficiate in analyzing the situation rigorously.

**Table 8**
Network forensics: challenges and proposed solutions.

| Network forensics challenges | Proposed solutions | Explanation | |
|---|---|---|---|
| Network speed | Specialized hardware e.g. NIFIC | – NIFIC: it contains of gigabit ethernet ports that capture high speed data packets, classify and filter, forward to stated interface, and perform packet analysis in FPGA programmable processing element. | Tripathi (2009) |
| | Software solution e.g. nCap library | 1. Capture packet between 1 and 10 gigabits speed<br>2. Ability to develop from user space<br>3. Use for active and passive monitoring of the network. | Deri (2005) |
| | Distributed packet capturing | 1. Capture packets with load balancing among several nodes<br>2. Cost effective due to no dedicated hardware required<br>3. Used simultaneously with other packets capturing tools | Morariu and Stiller (2008) |
| Storage capacity | Traffic archiving system with flow granularity | – TIFAFlow: Time machine based packet capturing, perform fast bit indexing and further store it on hard drive. It also increase flow query operation. | Chen et al. (2013b) |
| | Compress bitmap index in real time on GPU | 1. Store up to 185 million records per second<br>2. Indexing offloaded to GPU architecture<br>3. CPU intervention is scare | Fusco et al. (2013) |
| | Packet-to-disk application: n2disk | 1. Capture packet of any size in 10 gigabit at line rate on commodity hardware<br>2. Can be used for single thread and multi-thread packet consumers<br>3. Configurable to use in real time situation to index packet | Deri et al. (2013) |
| Data integrity | Systematic analysis using GUI-based monitoring | 1. Packets are judge by ensuring real time properties. This is performed by collecting servers, which further distribute analyzed information to the clients while also storing it in database<br>2. Performs hash function | Si-Young and Jong-Chan (2012) |
| Data privacy | Forensic attribution | 1. It helps investigator to view data of interest through forensic attribution<br>2. Each observer will verify packet signature whereas it enforce attribution property<br>3. Aforementioned can be achieved by using following methods (a) Group signatures, (b) BBS short group signatures | Afanasyev et al. (2011) |
| IP address problems | Source address validation | 1. Based on SAVI proposal<br>2. Binds source host IP, Mac addresses and uplink port properties in layer switches<br>3. No node can spoof IP addresses of attached node to same uplink | (Bi et al., 2013) |
| Data extraction locations | Central log repository | – Allow all network traffic to pass through central device installed for monitoring and analyzing. | Didier Stevens (2012) |
| | Targeting primary network devices | – This may be useful in single event of interest. But this might not provide complete evidence | |
| Intelligent network forensic tools | Fidelis XPS | 1. Capture, visualize, and record session of interest<br>2. Automatic response, reduce cost, increase bandwidth, and provide proactive awareness<br>3. Real time visualization | Savchuk (2013) |
| | WildPackets network forensic tools | 1. Capture, record, and analyze in 10 gigabit network traffic speed<br>2. Analyze data at point of capture in real time situation<br>3. Comprehensive data collection | McCreery (2012) |

### 5.3. Data integrity

Data integrity plays a vital role in the process of network forensics which has to be tackled (James, 2012). Data integrity in the network is an ability to keep accurate, complete, and consistent data in the network. Ascertaining integrity for data capturing on the network is a critical and challenging task for network forensics. The scope, size and velocity of data make it challenging for investigators to keep integrity of the data. Moreover, lack of the trust in data integrity itself makes data and data system unpredictable which further increases complexities for network investigators. Data integrity is affected by hardware and software errors, malicious attacks, malfunction of the system, and frequent mobility of the data present in the network. It adversely affects forensic process if the integrity of the data is not preserved upon being modified intentionally or deliberately. Data integrity incorporates security, reliability, and consistency, which are important factors for the network (Cusack and Alqahtani, 2013). Furthermore, data integrity assists network forensics investigators to mention the case in court to prosecute an intruder as criminals.

Data integrity needs to be solved with end-to-end mode, both software and hardware which has to be used in a seamless manner (Ahmed et al., 2013). Early alerts and remedial action taken to solve the problem is of utmost importance in today's fast growing networks. By creating methodology and standards, these aims can be realized in an efficient and cost effective manner in the realm of

data integrity especially in the distributed networks. A systematic analysis using GUI-based monitoring is proposed in which server analyze network packets in a real-time and distribute it towards client nodes for its storage with performing hash function (Si-Young and Jong-Chan, 2012). This enhances data integrity by analyzing packets with real time properties and stores such packets in different client storage to safe from various vulnerabilities.

However, the aim of data integrity is to ensure privacy and security in the forensic system legally. At any stage, data integrity should not be compromised in the entire forensic process. It is difficult to maintain integrity of the data when intruders perform their malicious round in the networks and alter their traces by using anti-forensics techniques (Lee et al., 2007, Whitteker, 2008). It creates hurdle for network forensic investigators to identify original evidence from the network and claim its integrity. In addition, other factor that contributes to minimizing data integrity is the volatile and dynamic nature of the network data itself. Millions of packets are transmitted in the data networks in no time due to the huge bandwidth and high data speed rate; which wipes out its appearance from the network on closing of the session. This incorporates complexity in the pool for data integrity especially in the distributed networks, which encompasses thousands of links that are challenging to be tackled.

### 5.4. Data privacy

Data privacy is an important factor in the investigation process of network forensics (Aminnezhad et al., 2012). A forensic attribution solution is proposed to solve the aforementioned problem related to user privacy (Afanasyev et al., 2011). A forensic investigator can view the data of interest by verifying the packet signature to enforce forensic attribution in the network. This can be achieved by using crypto-graphical tools, i.e. group signatures, and BBS short group signatures. If a group member creates a signature so it can be verified by the rest of the group members that signature has been created by our group member but cannot identify who has created it without cooperation with a third party group manager. Whereas in BBS short group signature exhibits relatively short signatures in comparison with group-signature schemes and is incorporated with its clue's optimization. Moreover, these crypto-graphical tools ensure that only authorized parties can determine the physical identity of the hardware originating from any given IP packets while the rest of it cannot recognize it. Consequently, it minimizes the problem of data privacy present in accessing distributed networks for retrieving digital evidence for the investigation.

Nonetheless, accessing data in the network of an organization might violate its privacy policies. Therefore, organizations become reluctant to allow network forensics investigators to perform their action to access important data. This is because a single trace file accessed in the network for malicious incident can contain other important records of multiple users or an organization, for instance financial records and employee records. Hence, organizations prefer not to permit any third party investigators to exploit their network data for any type of investigation. Furthermore, collecting network data raises different legal issues that could further delay the network forensics process including privacy and confidentiality of the user data. A trace file that is collected by an investigator can contain user password, email content, bank record, and its various other personal records. The sophistication increases for the network forensics when data is transmitted within various networks through different ISP's installed centrally. To collect network data from such ISPs, a legal request is initiated and could further delay the forensic process. It is further problematic if the investigation is demanded in real-time situation. As a result, ample network forensics technique is required to monitor and collect network traffic without any violation of user privacy and organization policies.

### 5.5. Access to IP addresses

The access of source IP address of an intruder is an important step in network forensics. Source IP address indicates origin of the attack (Fen et al., 2012a; Jeong and Lee, 2013) that assists in the identification of the intruder and stopping the attacks. Intruders employ different ways to hide their original source IP address from different network security devices installed in the network. For instance, it includes spoofing of IP address to result in the creation of forged source IP address with the purpose of hiding sender or with impersonating another person. It is mostly used in DDoS attacks to bombard network system with devastating amount of the traffic from different suspected systems. The spoofed IP address makes forensics investigation complicates in determining original source IP address especially in large distributed network environments (Heydari et al., 2010). To address the aforementioned issue, a source address validation improvement (SAVI) solution is proposed to bind source host IP address, mac address, and uplink port properties in layer switches in the network (Bi et al., 2013). It restricts attached nodes to the same uplink so that attackers cannot spoof IP addresses while staying in the same group of nodes. In addition, it uses anti-spoofing and traceback for IPv4/IPv6 transition situations by extracting crucial and common properties from several transition schemes. To conclude this section, this proposal can reduce IP spoofing which is performed by intruders to hide its source IP address and effect the system desperately.

Moreover, some intruders generate multiple sources IP addresses to obscure network forensics investigator in addressing them appropriately. The fake IP addresses are scattered in the network to generate further malicious packets from dispersed location resulting system overloaded; especially in DDoS attacks. Therefore, identification of correct source IP address to determine the origin of the attack is a challenge. In addition, most of systems in network are assigned dynamic IP addresses and it might have other IP addresses at the time of attack rather than having current IP address. This adds to the intricacy in validating the correct IP address for a right system at the right time. The aforementioned problems related to the IP address make system costly and more problematic when the investigation is requisite in real-time situation.

### 5.6. Data extraction location

Distributive nature and virtualized characteristics of networks complicate network forensics in identifying appropriate location and device for extracting the data. A network with thousands of devices connected with each other through high speed data links, which transmit millions of packets per second is difficult to be handled for its each link and device. Extracting data from appropriate location of large volume network for analyzing network packets is a radical challenge for the network forensics, which should not further compromise data privacy and integrity. Moreover, many devices are used to collect and extract data for network forensics including routers, IDS, firewalls, network forensics analysis tools, protocol analyzer, packet sniffers, and others. Furthermore, to place such devices at the right places to collect maximum level of data from the network for investigation and reconstructing attack paths is challenging. It is of great importance to determine right location, with a right device, and at right time to collect appropriate evidence from the network.

To overcome the problem of collecting data from the network, a centralized log repository is maintained by installing one or more devices to capture all network traffic which is diverted towards such centralized devices (Didier Stevens, 2012). Such an approach can be favorable in terms of monitoring and capturing network traffic but can be costly in terms of more memory space and congested bandwidth around such devices; which degrades the overall performance of the system. However, network traffic can be converted into group of similar events and is monitored and captured according to the specific event requirements. It generates an alert, when desired event packets are received and stored and analyzed centrally. Accordingly, it can solve the problem of storage but it uses more CPU cycles to perform such task in monitoring network traffic in terms of different events.

### 5.7. *Intelligent network forensic tools*

Current network forensic analysis tools capture and record network traffic by targeting complete packets. Such tools incorporate problems regarding storing huge volume of data with more time delays. An intelligent and smart network forensic tool is required to capture network traffic of choice depending on the investigational situation. For instance, capturing specific session data with a domain of interest, which further records, analyzes, and visualizes the data. This will reduce problem of storage, computational resources for investigation, bandwidth utilization, time delays, and result in quick incident response in real-time situation. The intelligent network forensic tool will give a complete visualize view of the network related to their ports, devices, channels, and protocols. Network forensics should now migrates towards intelligent network forensic tools to capture, record, and analyze network evidence from emerging network infrastructure including software defined networks (Akhunzada et al., 2015), cloud computing, mobile cloud computing (Shiraz et al., 2015), and now new evolving field, i.e. sky computing (Keahey et al., 2009).

An intelligent network forensic analysis tool, i.e. Fidelis XPS is introduced, which captures, visualizes, and records session of interest of the network traffic (Savchuk, 2013). It provides automatic response, reduced cost, increased bandwidth, proactive awareness, granular control, and results in real-time visualization. It restricts transmission of sensitive information by guarding malicious communications and payload based on the information gather from the compromised systems.

Data communication networks are growing fast and becoming more complex in data centers (Qi et al., 2014). Therefore, handling high speed data rate networks including 10 gigabits/s, 40 gigabits/s and now even 100 gigabits/s become intensive for single network management device. A comprehensive network forensics solution is required, to capture high speed data rate network traffic with having network analysis solution in place, to quickly identify and solve problems for ensuring safe and stable networks.

## 6. Conclusion and future directions

This paper reviews NFT used to identify and investigate legal evidence against intruders in the network infrastructure. Investigation aims at the origin of the attack, reliability and integrity of the evidence, visualization of attack paths, and determining worst attack paths. The investigation goals are achievable whenever investigators are clear about the network infrastructure and attack behavior by having appropriate network forensic tools and extensive network forensics knowledge. Moreover, NFT play a vital role in identifying, capturing, recording, and analyzing legal evidences in distributed networks. The NFT are required to be scalable with increasing network infrastructure in order to analyze fast moving and huge amount of network packets collected at various locations in the network. A number of current NFT involve the problem of storage which can be solved by capturing relevant infected network packets. This requires smart and efficient data reduction and filtering techniques to reduce storage and computational requirements for sheer amount of the network traffic. Conversely, a comprehensive solution is desired in deploying, managing, and bearing less cost for network forensics strategies in distributed networks. As a result, network security and its visibility in network complexity will be improved and managed easily. Therefore, network forensics solution and legal standards demand further research to ensure the availability of real-time network traffic by identifying suspicious attacks, internal security violation, and inappropriate use of network resources in the network.

The development of intelligent network forensic tools to focus on specific type of network traffic analysis is a challenge in terms of future perspective. This will reduce time delays, less computational resources requirement; minimize attacks, providing reliable and secured evidences, and efficient investigation with minimum efforts. Moreover, network forensics at distributed networks of the cloud computing and mobile cloud computing needs to be explored (Khan et al., 2016b). It increases the sophistication of network forensics investigation due to the frequent migration of data, virtualized networking, and multi-tenant support; inter cloud network connectivity, and trillions of resources in the cloud environment. In conclusion, a standard legal procedure is required at present to help network forensics investigators in having access to the network devices at various network positions in the cloud such as cloud access network, data center network, and inter cloud network.

## Acknowledgments

This work is fully funded by Bright Spark Unit, University of Malaya, Malaysia and partially funded by the High Impact Research Grant UM.C/625/1/HIR/MOE/FCSIT/17, Malaysian Ministry of Higher Education under the University of Malaya.

## References

Afanasyev M, Kohno T, Ma J, Murphy N, Savage S, Snoeren AC, et al. Privacy-preserving network forensics. Commun. ACM 2011;54:78–87.

Ahmed E, Khan, S., Yaqoob, I., Gani, A., Salim, F., 2013. Multi-objective optimization model for seamless application execution in mobile cloud computing. Information and Communication Technologies (ICICT). In: Procedings of the 5th International Conference on: IEEE, pp. 1–6.

Akhunzada A, Gani A, Anuar NB, Abdelaziz A, Khan MK, Hayat A, et al. Secure and dependable software defined networks. J. Netw. Comput. Appl. 2015.

Akyuz, T., Sogukpinar, I., 2009. Packet marking with distance based probabilities for IP traceback. In: Proceedings of the NETCOM'09, First International Conference on Networks and Communications, IEEE, pp. 433–438.

Albanese M, Jajodia S, Pugliese A, Subrahmanian V. Scalable Analysis of Attack Scenarios. Computer Security–ESORICS 2011. In: Berlin, Germany: Springer; 2011. p. 416–33.

Alharbi S, Weber-Jahnke J, Traore I. The Proactive and Reactive Digital Forensics Investigation Process: A Systematic Literature Review. Information Security and Assurance. Berlin, Germany: Springer; 87–100.

Aminnezhad A, Dehghantanha A, Abdullah MT. A survey on privacy issues in digital forensics. Int. J. Cyber-Secur. Digit. Forensics (IJCSDF)) 2012;1:311–23.

Anderson, J., 2001. An Analysis of Fragmentation Attacks. Unpublished manuscript, available at: ⟨http://wwwouah org/fragma html⟩.

Anwar, S., Zain, J.M., Zolkipli, M.F., Inayat, Z., Jabir, A.N., Odili, J.B., 2015. Response option for attacks detected by Intrusion Detection System. In: Proceedings of the 4th International Conference on Software Engineering and Computer Systems (ICSECS), IEEE, pp. 195–200.

Bejtlich, R., 2013. The Practice of Network Security Monitoring: Understanding Incident Detection and Response. No Starch Press.

Belenky, A., Ansari, N., 2003. IP Traceback with deterministic packet marking, Commun. Lett., IEEE, 7, pp. 162–164.

Belenky A, Ansari N. On deterministic packet marking. Comput. Netw. 2007;51:2677–700.

Bi, J., Deng, H., Xu, M., Shi, F., Hu, G., 2013. A General Framework of Source Address Validation and Traceback for IPv4/IPv6 Transition Scenarios.

Bo, F., Fan, G., Min, Y., 2009. Dynamic probabilistic packet marking based on PPM. In: Proceedings of the WMWA'09s Pacific-Asia Conference on Web Mining and Web-based Application, IEEE, pp. 289–292.

Boruah, A., Hazarika, S.M., 2014. An MEBN framework as a dynamic firewall's knowledge flow architecture. In: Proceedings of the International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. 249–254.

Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput. Surv. 2009;41:15.

Chen, L., Li, Z., Gao, C., Liu, Y., 2009. Modeling and analyzing dynamic forensics system based on intrusion tolerance. In: Proceedings of the CIT'09 Ninth IEEE International Conference on Computer and Information Technology, IEEE, pp. 230–235.

Chen LM, Chen MC, Liao W, Sun YS. A scalable network forensics mechanism for stealthy self-propagating attacks. Comput. Commun. 2013a.

Chen S, Tang Y, Du W. Stateful DDoS attacks and targeted filtering. J. Netw. Comput. Appl. 2007;30:823–40.

Chen Z, Ruan L, Cao J, Yu Y, Jiang X. TIFAflow: enhancing traffic archiving system with flow granularity for forensic analysis in network security. Tsinghua Sci. Technol. 2013b:18.

Cheng B-C, Liao G-T, Huang H-C, Hsu P-H. Cheetah: a space-efficient HNB-based NFAT approach to supporting network forensics. Ann. Telecommun. 2013:1–11.

Clegg, R.G., Withall, M.S., Moore, A.W., Phillips, I.W., Parish, D.J., Rio, M., et al., 2013. Challenges in the Capture and Dissemination of Measurements From High-Speed Networks. arXiv preprint arXiv:13036908.

Cordon O, Herrera F, Stützle T. A review on the ant colony optimization metaheuristic: basis, models and new trends. Mathw. Soft Comput.: Citeseer 2002.

Cusack, B., Alqahtani, M., 2013. Acquisition of Evidence From Network Intrusion Detection Systems.

Danforth, M., 2006. Models for threat assessment in networks. DTIC Document.

Deri, L., 2005. nCap: Wire-speed packet capture and transmission. End-to-End Monitoring Techniques and Services 2005 Workshop on: IEEE, pp. 47–55.

Deri, L., Cardigliano, A., Fusco, F., 2013. 10 Gbit line rate packet-to-disk using n2disk. In: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, pp. 441–446.

Diamah A, Mohammadian M, Balachandran BM. Network Security Evaluation Method via Attack Graphs and Fuzzy Cognitive Maps. Intelligent Decision Technologies. Berlin, Germany: Springer; 433–40.

Didier Stevens Network Device Forensics ISSA, 2012.

Domingos, P., Hulten, G., 2000. Mining high-speed data streams. In: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 71–80.

Dou W, Chen Q, Chen J. A confidence-based filtering method for DDoS attack defense in cloud environment. Future Gener. Comput. Syst. 2012.

Ebrahimi, A. , Navin, A., Mirnia, M., Bahrbegi, H., Ahrabi, A., 2011. Automatic Attack Scenario Discovering Based on a New Alert Correlation Method Systems Conference (SysCon), IEEE International, IEEE, pp. 52–58.

Emre Y. A Literature Survey About Recent Botnet Trends. 2011.

Fan, Y.-T., Wang, S.-J., 2010. Intrusion investigations with data-hiding for computer log-file forensics future information technology (FutureTech). In: Proceedings of the 5th International Conference on: IEEE, pp. 1–6.

Fen Y, Hui Z, Shuang-shuang C, Xin-chun Y. A lightweight IP traceback scheme depending on TTL. Procedia Eng. 2012a;29:1932–7.

Fen Y, Xinchun Y, Hao H. An network attack modeling method based on MLL-AT. Phys. Procedia 2012b;24:1765–72.

Fernandez E, Pelaez J, Larrondo-Petrie M. Attack Patterns: A New Forensic and Design Tool. Advances in Digital Forensics III. Berlin, Germany: Springer; 345–57.

Forte DV. The “art” of log correlation: tools and techniques for correlating events and log files. Comput. Fraud Secur. 2004:15–7.

Fusco, F., Vlachos, M., Dimitropoulos, X., Deri, L., 2013. Indexing million of packets per second using GPUs. In: Proceedings of the Conference on Internet Measurement Conference, ACM, pp. 327–332.

Gani A, Nayeem GM, Shiraz M, Sookhak M, Whaiduzzaman Md, Khan S. A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing. J. Netw. Comput. Appl. 2014.

Garfinkel SL. Digital forensics research: the next 10 years. Digit. Investig. 2010;7:S64–73.

Gong, C., Sarac, K., 2005. IP traceback based on packet marking and logging. IEEE International Conference on Communications 2005 ICC 2005, IEEE, pp. 1043–1047.

Goralski W. The Illustrated Network: How Tcp/ip Works in a Modern Network. Burlington: Morgan Kaufmann; 2009.

Grobler, C., Louwrens, C., von Solms, S.H., 2010. A multi-component view of digital forensics. In: Proceedings of the ARES'10 International Conference on Availability, Reliability, and Security, IEEE, pp. 647–652.

Harbort, Z., Louthan, G., Hale, J., 2011. Techniques for attack graph visualization and interaction. In: Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, ACM, p. 74.

Heydari, S., Martin, M.V., Rjaibi, W., Lin, X., 2010. Emerging trends in network forensics. In: Proceedings of the Conference of the Center for Advanced Studies on Collaborative Research, IBM Corp., pp. 389–390.

Hoque N, Bhuyan MH, Baishya RC, Bhattacharyya DK, Kalita JK. Network attacks: taxonomy, tools and systems. J. Netw. Comput. Appl. 2014;40:307–24.

Hsiao, H.-W., Chen, D.-N., Wu, T.J., 2010. Detecting hiding malicious website using network traffic mining approach. In: Proceedings of the 2nd International Conference on Education Technology and Computer (ICETC), IEEE, p. V5-276-V5-80.

Hsu H-M, Sun YS, Chen MC. Collaborative scheme for VoIP traceback. Digit. Investig. 2011;7:185–95.

Ibrahim, M., Abdullah, M.T., Dehghantanha, A., 2012. VoIP evidence model: A new forensic method for investigating VoIP malicious attacks. Cyber Warfare and Digital Forensic (CyberSec) International Conference on Cyber Security, IEEE, pp. 201–206.

Igure V, Williams R. Taxonomies of attacks and vulnerabilities in computer systems. Commun. Surv. Tutor. IEEE 2008;10:6–19.

Inayat Z, Gani A, Anuar NB, Khan MK, Anwar S. Intrusion response systems: foundations, design, and challenges. J. Netw. Comput. Appl. 2015.

Ingols, K., Lippmann, R., Piwowarski, K., 2006. Practical attack graph generation for network defense. Computer Security Applications Conference, ACSAC'06 22nd Annual, IEEE, pp. 121–130.

James PJ. The Importance Of Network Security Forensics. 2012.

Jemili F, Zaghdoud M, Ben Ahmed M. A framework for an adaptive intrusion detection system using Bayesian network. Intelligence and Security Informatics, 2007 IEEE, IEEE; 2007. p. 66-70.

Jeong E, Lee B. An IP traceback protocol using a compressed hash table, a sinkhole router and data mining based on network forensics against network attacks. Future Gener. Comput. Syst. 2013.

Jiang J, Ma H, Ren D. Network security intrusion detection. J. Softw. 2000.

Jiang, L., Tian, G., Zhu, S., 2012. Design and implementation of network forensic system based on intrusion detection analysis. In: Proceedings of the International Conference on Control Engineering and Communication Technology (ICCECT), IEEE, pp. 689–692.

Jin, C., Wang, X.-Y., Tan, H.-Y., 2010. Dynamic attack tree and its applications on trojan horse detection. In: Proceedings of the Second International Conference on Multimedia and Information Technology (MMIT), IEEE, pp. 56–59.

Jin G, Yang J. Deterministic packet marking based on redundant decomposition for IP traceback. Commun. Lett. IEEE 2006;10:204–6.

Johnston, A.B., Piscitello, D.M., 2006. Understanding Voice over Ip Security (Artech House Telecommunications Library), Artech House, Inc.

Kailiponi P. Analyzing evacuation decisions using multi-attribute utility theory (MAUT). Procedia Eng. 2010;3:163–74.

Keahey K, Tsugawa M, Matsunaga A, Fortes J. Sky computing. Int. Comput. IEEE 2009;13:43–51.

Kent K, Souppaya M. Guide to Computer Security Log Management. NIST Special Publication; 800–92.

Khan S, Gani A, Abdul Wahab AW, Abdelaziz A, Bagiwa MA, FML: A novel Forensics Management Layer for Software Defined Networks. In 6th International Conference on Cloud System and Big data Engineering, Confluence-2016, 14-15 Jan, 2016a, Amity University, Noida, UP India. (In Press).

Khan S, Gani A, Abdul Wahab AW, Shiraz M, Bagiwa MA, Khan SU, Buyya RK, Zomaya AY, Cloud Log Forensics: Foundations, State-of-the-art, and Future Directions, ACM Computing Surveys. 2016b. (In Press).

Khan, S., Ahmad, E., Shiraz, M., Gani, A., Wahab, A.W.A., Bagiwa, M.A., 2014a. Forensic challenges in mobile cloud computing. In: Proceedings of the International Conference on Computer, Communications, and Control Technology (I4CT), IEEE, pp. 343–347.

Khan, S., Gani, A., Wahab, A.W.A., Bagiwa, M.A., 2015. SIDNFF: Source identification network forensics framework for cloud computing. In: Proceedings of the IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), IEEE, pp. 418–419.

Khan S, Hayat K, Madani SA, Khan SU, Kolodziej J. The median resource failure checkpointing. ECMS 2012:483–9.

Khan S, Shiraz M, Abdul Wahab AW, Gani A, Han Q, Bin Abdul Rahman Z. A comprehensive review on adaptability of network forensics frameworks for mobile cloud computing. Sci. World J. 2014b:2014.

Kim, H.S., Kim, H.K., 2011. Network Forensic Evidence Acquisition (NFEA) with packet marking. In: Proceedings of the Ninth IEEE International Symposium on Parallel and Distributed Processing with Applications Workshops (ISPAW), IEEE, pp. 388–393.

Kok, J., Kurz, B., 2011. Analysis of the BotNet ecosystem. In: Proceedings of the 10th Conference on Telecommunication, Media and Internet Techno-Economics (CTTE), VDE, pp. 1–10.

Kruegel, C., Mutz, D., Robertson, W., Valeur, F., 2003. Bayesian event classification for intrusion detection. In: Proceedings of the19th Annual Computer Security Applications Conference, IEEE, pp. 14–23.

Layton, R., Watters, P., Dazeley, R., 2010. Automatically determining phishing campaigns using the USCAP methodology. eCrime Researchers Summit (eCrime), IEEE, pp. 1–8.

Lee, S.S., Chang, K.-Y., Lee, D., 2007. A new anti-forensic tool based on a simple data encryption scheme. Future Generation Communication and Networking (FGCN 2007), IEEE, pp. 114–118.

Levine BN, Shields C. Hordes: a multicast based protocol for anonymity. J. Comput. Secur. 2002;10:213–40.

Li, J., Sung, M., Xu, J., Li, L., 2004. Large-scale IP traceback in high-speed Internet: Practical techniques and theoretical foundation. In: Proceedings of the 2004 IEEE Symposium on Security and Privacy, IEEE, pp. 115–129.

Li P, Salour M, Su X. A survey of internet worm detection and containment. Commun. Surv. Tutor. IEEE 2008;10:20–35.

Li S, Schmitz R. A Novel Anti-Phishing Framework Based on Honeypots. IEEE; 2009.
Liao H-J, Richard Lin C-H, Lin Y-C, Tung K-Y. Intrusion detection system: a comprehensive review. J. Netw. Comput. Appl. 2013;36:16–24.
Lin, I.-L., Yen, Y.-S., Wu, B.-L., Wang, H.-Y., 2010. VoIP network forensic analysis with digital evidence procedure. In: Proceedings of the Sixth International Conference on Networked Computing and Advanced Information Management (NCM), IEEE, pp. 236–241.
Liu, C., Singhal, A., Wijesekera, D., 2012. Using Attack Graphs in Forensic Examinations. In: Proceedings of the Seventh International Conference on Availability, Reliability and Security (ARES), IEEE, pp. 596–603.
Liu, X., Fang, C., Xiao, D., Xu, H., 2010. A goal-oriented approach for modeling and analyzing attack graph. In: Proceedings of the International Conference on Information Science and Applications (ICISA), IEEE, pp. 1–8.
Liu, Y., 2010. Study on application of apriori algorithm in data mining. In: Proceedings of the ICCMS'10 Second International Conference on Computer Modeling and Simulation, IEEE, pp. 111–114.
Ma, J., Wang, Y., Sun, J., Hu, X., 2010. A Scalable, Bidirectional-Based Search Strategy to Generate Attack Graphs. In: Proceedings of the IEEE 10th International Conference on Computer and Information Technology (CIT), IEEE, pp. 2976–2981.
McCreery TD. Network Forensics in a 10G World WildPackets. 2012.
Mitropoulos, S., Patsos, D., Douligeris, C., 2005. Network forensics: towards a classification of traceback mechanisms. In: Proceedings of the Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, IEEE, pp. 9–16.
Mizoguchi, S., Takemori, K., Miyake, Y., Hori, Y., Sakurai, K., 2011. Traceback Framework against Botmaster by Sharing Network Communication Pattern Information. In: Proceedings of the Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), IEEE, pp. 639–644.
Morariu, C., Stiller, B., 2008. DiCAP: Distributed Packet Capturing architecture for high-speed network links. In: Proceedings of the 33rd IEEE Conference on Local Computer Networks, 2008 LCN, IEEE, pp. 168–175.
Mosa, M.R., Jantan, A., 2013. A New Algorithm to Estimate the Similarity between the Intentions of the Cyber Crimes for Network Forensics. In: Proceedings of the 4th International Conference on Electrical Engineering and Informatics.
Nagesh A. Distributed Network Forensics Using JADE Mobile Agent Framework. Tempe, USA: Arizona State University; 2006.
Nelson, B., Phillips, A., Steuart, C., 2010. Guide to Computer Forensics and Investigations. CengageBrain.com.
NIST, 2002. Federal Information Security Management Act of 2002.
Noh S, Lee C, Choi K, Jung G. Detecting Distributed Denial of Service (DDoS) Attacks Through Inductive Learning. Intelligent Data Engineering and Automated Learning. Berlin, Germany: Springer; 286–95.
Orebaugh A. Proactive forensics. J. digit. Forensic Pract. 2006;1:37–41.
Ou, X., Govindavajhala, S., Appel, A.W., 2005. MulVAL: A logic-based network security analyzer. In: Proceedings of the 14th USENIX Security Symposium, pp. 1–16.
Palmer, G., 2001. A Road Map for Digital Forensics Research-report From the first Digital Forensics Research Workshop (DFRWS), Utica, New York.
Park P. VoicE Over IP Security. Pearson Education; 2008.
Pelaez J.C., Fernandez E.B., Voip network forensic patterns. Computing in the Global Information Technology, 2009 ICCGI'09. In: Proceedings of the Fourth International Multi-Conference on: IEEE, pp. 175–180.
Perdisci, R., Lee, W., Feamster, N., 2010. Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces. NSD. pp. 391–404.
Perry S. Network forensics and the inside job. Netw. Secur. 2006;2006:11–3.
Pilli ES, Joshi RC, Niyogi R. Network forensic frameworks: survey and research challenges. Digit. Investig. 2010;7:14–27.
Ponec M, Giura P, Wein J, Brönnimann H. New payload attribution methods for network forensic investigations. ACM Trans. Inf. Syst. Secur. (TISSEC) 2010;13:15.
Qi H, Shiraz M, Gani A, Whaiduzzaman M, Khan S. Sierpinski triangle based data center architecture in cloud computing. J. Supercomput. 2014;69:887–907.
Qu, Z., Huang, C., 2008. A novel deterministic packet marking scheme for IP traceback. In: Proceedings of the PEITS'08 Workshop on Power Electronics and Intelligent Transportation System, IEEE, pp. 38–41.
Raftopoulos, E., Dimitropoulos, X., 2013. Understanding Network Forensics Analysis in an Operational Environment. Security and Privacy Workshops (SPW) IEEE, IEEE, pp. 111–118.
Ray, I., Poolsapassit, N., 2005. Using Attack Trees to Identify Malicious Attacks From Authorized Insiders. Computer Security–ESORICS 2005, Springer pp. 231–246.
Rekhis, S., 2007. Theoretical Aspects of Digital Investigation of Security Incidents: Citeseer.
Ren, W., 2004. On a Reference Model of Distributed Cooperative Network, Forensics System. iiWAS.
Ren, W., Jin, H., 2005. Distributed agent-based real time network intrusion forensics system architecture design. In: Proceedings of the 19th International Conference on Advanced Information Networking and Applications, AINA 2005, IEEE, pp. 177–182.
Roesch M. Snort: Lightweight Intrusion Detection for Networks. LISA; 229–38.
Ruia, A., Parekh, V., Chakrabarti, A., 2011. An indexing method for efficient querying of an attack graph. In: Proceedings of the International Conference on Recent Trends in Information Systems (ReTIS), IEEE, pp. 82–87.
Saha, D., 2008. Extending logical attack graphs for efficient vulnerability analysis. In: Proceedings of the 15th ACM conference on Computer and communications security, ACM, pp. 63–74.
Sanguankotchakorn, T., Dechasawatwong, T., 2011. Automatic attack detection and correction system development. In: Proceedings of the 13th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, pp. 1–8.
Savchuk, A., 2013. Intelligent Network Forensics. Fidelis Cybersecurity Solutions.
Services USDoHH, 1996. Summary of the HIPAA Security Rule.
Shanmugasundaram K, Memon N, Savant A, Bronnimann H. ForNet: A distributed forensics network. Computer Network Security. Springer; 1–16.
Sheyner, O., Haines, J., Jha, S., Lippmann, R., Wing, J.M., 2002. Automated generation and analysis of attack graphs. I: Proceedings of the 2002 IEEE Symposium on Security and Privacy, IEEE, pp. 273–284.
Shiraz M, Gani A, Shamim A, Khan S, Ahmad RW. Energy efficient computational offloading framework for mobile cloud computing. J.Grid Comput. 2015;13:1–18.
Si-Young N, Jong-Chan L. Forensic mechanism for data integrity. Smart Comput. Rev. 2012;2.
Siris VA, Stavrakis I. Provider-based deterministic packet marking against distributed DoS attacks. J. Netw. Comput. Appl. 2007;30:858–76.
Smith AD. Exploring security and comfort issues associated with online banking. Int. J. Electron. Financ. 2006;1:18–48.
Sperotto A, Schaffrath G, Sadre R, Morariu C, Pras A, Stiller B. An overview of IP flow-based intrusion detection. Commun. Surv. Tutor. IEEE 2010;12:343–56.
Srinivasan, R.S.B.B.A., P5: A Protocol for Scalable Anonymous Communication IEEE Secur. Priv. 2002.
Sullivan, B., 2011. Application-Level Denial of Service Attacks and Defenses Presented in conjunction with the BlackHat DC 2011.
Sy BK. Integrating intrusion alert information to aid forensic explanation: an analytical intrusion detection framework for distributive IDS. Inf. Fus. 2009;10:325–41.
Takemori, K., Fujinaga, M., Sayama, T., Nishigaki, M., 2008. IP traceback using DNS logs against bots.2008 CSA'08 International Symposium on Computer Science and its Applications: IEEE. pp. 84–89.
Thapliyal, M., Bijalwan, A., Garg, N., Pilli, E.S., 2013. A Generic Process Model for Botnet Forensic Analysis.
Tripathi, P., 2009. Packet Capturing Challenges and Approaches.
Wang, C., Feng, T., Kim, J., Wang, G., Zhang, W., 2009. Catching packet droppers and modifiers in wireless sensor networks. In: Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, 2009 SECON'09, IEEE, pp. 1–9.
Wang, D., Li, T., Liu, S., Zhang, J., Liu, C., 2007. Dynamical network forensics based on immune agent. In: Proceedings of the Third International Conference on Natural Computation, 2007 ICNC, IEEE, pp. 651–656.
Wang, F., Upppalli, R., 2003. Sitar: a scalable intrusion-tolerant architecture for distributed services-a technology summary. DARPA Information Survivability Conference and Exposition, 2003 Proceedings, IEEE, pp. 153–155.
Wang W, Daniels TE. A graph based approach toward network forensics analysis. ACM Trans. Inf. Syst. Secur. 2008;12:1–33.
Wang, W., Wei, S., Liao, Q., 2008. Classifying Infrasound with Wavelet Networks. 2008 CISP'08 Congress on Image and Signal Processing, IEEE, pp. 33–37.
Wang X-J, Wang X-Y. Topology-assisted deterministic packet marking for IP traceback. J. China Univ. Posts Telecommun. 2010;17:116–21.
Whitley JN, Phan RC-W, Wang J, Parish DJ. Attribution of attack trees. Comput. Electr. Eng. 2011;37:624–8.
Whitteker M. Anti-forensics: breaking the forensic process. Inf. Syst. Secur. Assoc. J. 2008:10–6.
Xiao-jing, W., Sheng-jun, W., 2009. IP traceback based probabilistic packet marking and randomized network coding. In: Proceedings of the Second International Workshop on Computer Science and Engineering, 2009 WCSE'09, IEEE, pp. 151–154.
Xie, Y., Sekar, V., Maltz, D.A., Reiter, M.K., Zhang, H., 2005. Worm origin identification using random moonwalks. IEEE Symposium on Security and Privacy, 2005. IEEE, pp. 242–256.
Yaar, A., Perrig, A., Song, D., 2005. FIT: fast Internet traceback. INFOCOM 2005. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE, IEEE, pp. 1395–1406.
Yang C-H, Weng C-Y, Wang S-J, Sun H-M. Adaptive data hiding in edge areas of images with spatial LSB domain systems. IEEE Tran. Inf. Forensics Secur. 2008;3:488–97.
Yonghui, L., Yulong, W., Fangchun, Y., Sen, S., Dong, Y., 2010. Deterministic packet marking based on the coordination of border gateways. In: Proceedings of the 2nd International Conference on Education Technology and Computer (ICETC), IEEE, p. V2-154-V2 – 61.
Yu B, Wang R. Research of Access Control List in Enterprise Network Management. Informatics and Management Science VI. In: Berlin, Germany: Springer; 2013. p. 121–9.
Yu, W., Fu, X., Blasch, E., Pham, K., Shen, D., Chen, G. et al., 2013. On effectiveness of hopping-based spread spectrum techniques for network forensic traceback. In: Proceedings of the 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), IEEE, pp. 101–106.

Zander S, Armitage GJ, Branch P. A survey of covert channels and countermeasures in computer network protocols. IEEE Commun. Surv. Tutor. 2007;9:44–57.
Zhang, F., Wang, D., 2013. An Effective Feature Selection Approach for Network Intrusion Detection. In: Proceedings of the IEEE Eighth International Conference on Networking, Architecture and Storage (NAS), IEEE, pp. 307–311.
Zhang Y, Jiang B, Wang J. Design of network forensic system based on intrusion tolerance. Comput. Eng. 2007;33:161–3.
Zhang Z, Wang S, Kadobayashi Y. Exploring attack graph for cost-benefit security hardening: a probabilistic approach. Comput. Secur. 2012.
Zhu Y. Attack pattern discovery in forensic investigation of network attacks. IEEE J. Sel. Areas Commun. 2011;29:1349–57.