Chapter 3
Network Forensic Frameworks

Learning Objectives

- Background of various process models in digital forensics
- Understanding of various phases in different process models
- Study of proposed models specific for network forensics
- Discussion on a generic process model for network forensics

The Network forensic process models were introduced in the previous chapter. The term 'model' has been used to imply a theoretical representation of phases involved in network forensics. This model may or may not have been implemented. A generic process model for network forensics was also discussed. The model considered only phases applicable to networked environments, based on the existing models of digital forensics.

Network forensic frameworks are surveyed in this chapter. Some of the models discussed earlier are implemented. The term 'framework' is used to mean practical implementation. These frameworks have been categorized into seven categories based on the technology used to build network forensic framework – distributed systems, soft computing, honeypots, attack graphs, data mining and aggregation systems. This chapter gives an overview of various techniques which can be used to build new frameworks and tools for network forensics.

3.1 Distributed Systems-Based Frameworks

Nowadays, network is spread all over the world through wired or wireless technology. Network forensic frameworks are described which are distributed in nature and help in understanding how log files and useful data can be extracted from various locations in the network.

Sundaram et al. [1] propose ForNet, a distributed network logging mechanism to aid digital forensics over wide area networks. It has two functional components – a

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_3

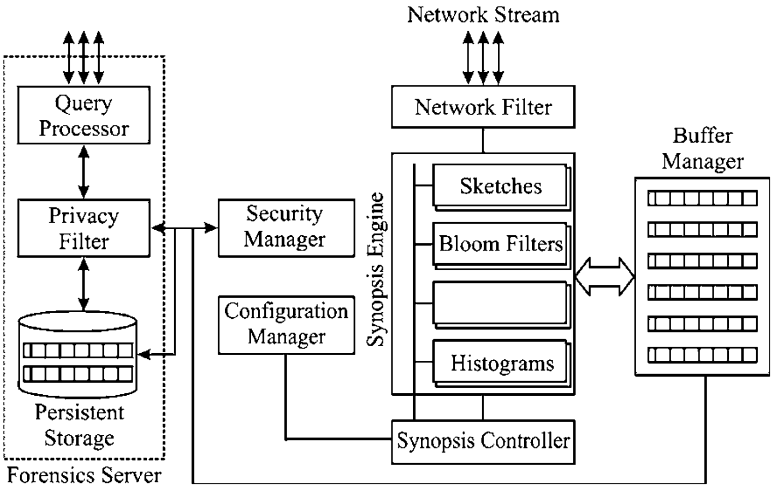

Fig. 3.1 Architecture of a SynApp with an integrated forensic server

SynApp, designed to summarize and remember network events for a period of time, and a *Forensic Server*, which is a centralized authority for a domain that manages a set of SynApps in that domain. A Forensic Server receives queries from outside its domain, processes them in cooperation with the SynApps, and returns query results back to the senders after authentication and certification. The overall architecture involves a network filter, Synopsis Engine, Synopsis Controller, Configuration Manager, Security Manager, Storage Management, and Query Processor. Evidence of crimes can be found in packet headers or application-dependent payloads. ForNet can identify network events like TCP connection establishment, port scanning, and connection record details and use bloom filters to track other events. The model is represented in Fig. 3.1.

Wei [2] proposed a reference model of distributed cooperative network forensic system. It is based on client-server architecture. The server captures network traffic, builds mapping topology database, filters, dumps, and transforms the network traffic stream into database values, mines forensic database, and replays network behavior. It also does network surveying, attack statistic analysis, and visualization. The distributed agent clients integrate data from firewall, IDS, honeynet, and remote traffic. The goal of this model is dumping the misbehavior packets traffic on the basis of adaptive filter rules, analyzing the overall cooperative database to discover the potential misbehavior, and replaying the misbehavior for the analysis of forensics. It can discover the profile of the attacker and obtain clues for further investigation. Proposed system is shown in Fig. 3.2.

Wei and Jing [3] further extended the above model as distributed agent-based real-time network intrusion forensic system. The goals of this framework include log system information gathering, adaptive capture of network traffic, active response for investigational forensics, integration of forensic data, and storing the historical network misuse pattern. The four elements in the system are network

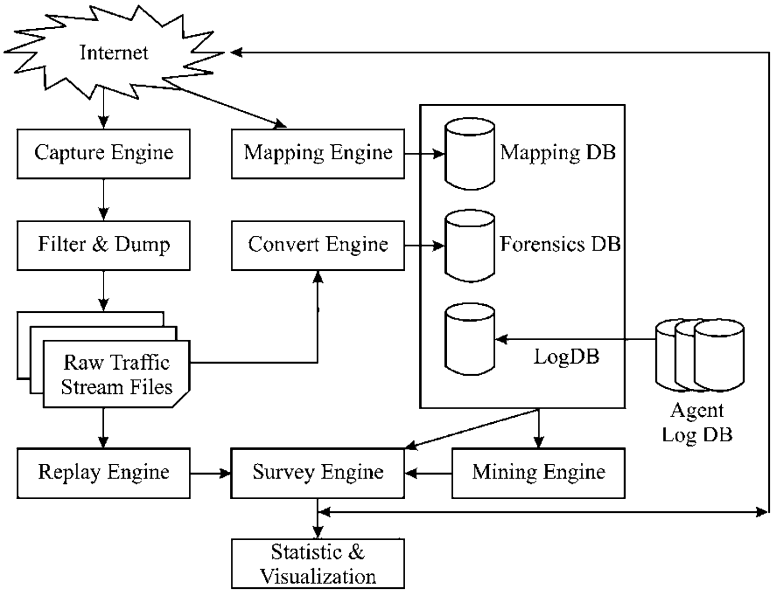

Fig. 3.2 The architecture of the network forensic server

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

Tang et al. [4] proposed a simple framework for distributed forensics. It's based on distributed techniques providing an integrated platform for automatic evidence collection and efficient data storage, easy integration of known attribution methods, and an attack attribution graph generation mechanism. The model is based on proxy and agent architecture. Agents collect, store, reduce, process, and analyze data. Proxies generate the attack attribution graph and perform stepping-stone analysis. This model aims at providing a method to collect, store, and analyze forensic information. It also provides automatic evidence and quick response to attacks. The model is represented in Fig. 3.3.

Nagesh [5] implemented a distributed network forensic framework using JADE mobile agent architecture. A node acting as a server, hosting the network forensic agent, dispatches mobile agents to monitored heterogeneous locations. They gather network traffic logs, examine them, and return the results which will be displayed on a user interface. The interface enables the analyst to specify the data to be collected and analyze the resultant network events displayed. The solution automates

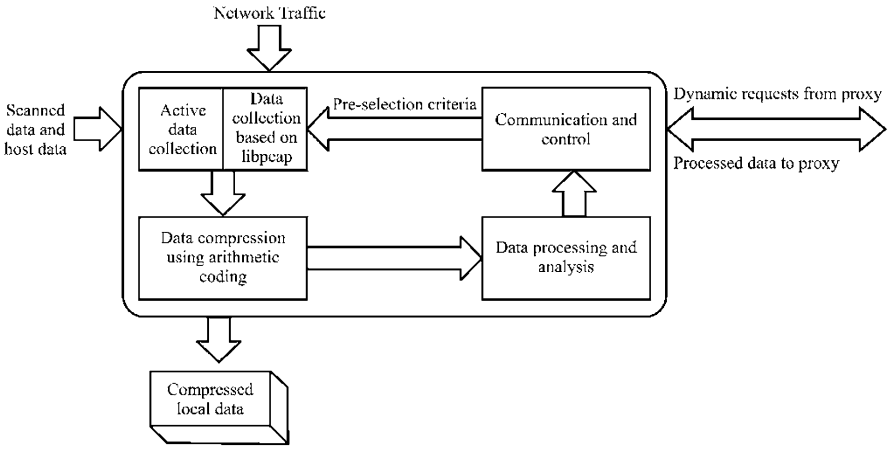

Fig. 3.3 Structure of agent

collection of network data from distributed heterogeneous systems using mobile agents, and the implementation is scalable, reduces network traffic, addresses a single point of failure, and provides real-time monitoring.

Wang et al. [6] developed a dynamical network forensic (DNF) model based on artificial immune theory and multi-agent theory. The system provides a real-time method to collect, stores the data logs simultaneously, and provides automatic evidence collection and quick response to network criminals. The system includes a Forensic Server and three agents, namely, detector agent, forensic agent, and response agent. Detector agent captures real-time network data, matches it with intrusion behavior, and sends a forensic request message to the forensic agent. The forensic agent collects the digital evidence, creates a digital signature using a hash function, and transmits the evidence to the Forensic Server. The Forensic Server analyzes the evidence and replays the attack procedure. The response agent is being developed.

3.2 Soft Computing-Based Frameworks

Soft computing technique helps in digital forensic to analyze the data and detect digital evidences automatically. It also helps to decrease the volume of data. In this section, we have discussed many soft computing methods which are used in network forensics, such as fuzzy logic, neuro-fuzzy, artificial neural network.

Kim et al. [7] develop a fuzzy logic-based expert system for network forensics to aid the decision-making processes involving sources of imprecision that are nonstatistical in nature. The system (shown in Fig. 3.4) can analyze computer crime in networked environments and make digital evidences automatically. It can provide

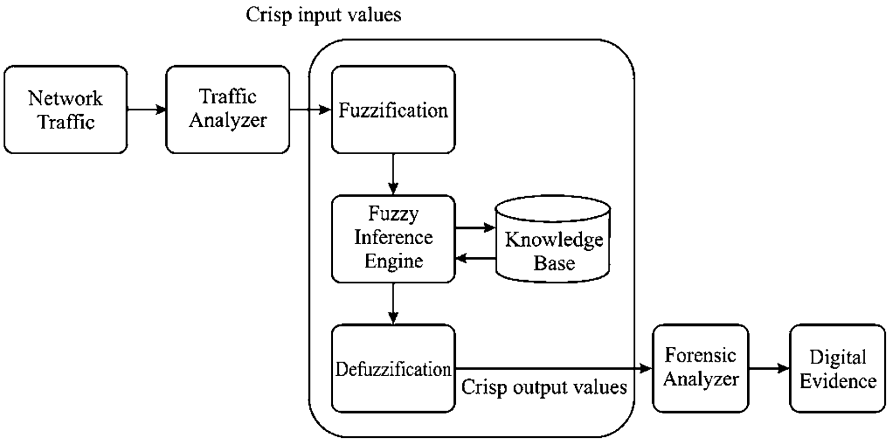

Fig. 3.4 The architecture of the fuzzy expert system for network forensics

analyzed information for a forensic expert and reduce the time and cost of forensic analysis. The framework consists of six components. Traffic analyzer captures network traffic and analyzes the same using sessionizing. Knowledge base stores the rules which are used by the fuzzy inference engine. The rules are written for various attacks using linguistic variables and terms. Membership functions are defined for each fuzzy set, and a crisp value of degree of membership is determined. Each input variables crisp value is first fuzzified into linguistic values. Fuzzy inference engine derives output linguistic values using aggregation and composition. Defuzzification defuzzifies the outvalues into crisp values, and the Forensic Analyzer decides whether captured packets indicate an attack.

Liu et al. [8] proposed the Incremental Fuzzy Decision Tree-Based Network Forensic System (IFDTNFS), which is shown in Fig. 3.5. This is an efficient way to create a classification model by extracting key features from network traffic by providing the resulting fuzzy decision tree with better noise immunity and increasing applicability in uncertain or inexact contexts. IFDTNFS consists of three components: Network Traffic Separator, Traffic Detector, and Forensic Analyzer. The Network Traffic Separator component is responsible for capturing network traffic and separating the network traffic according to the service type and directing the separated traffic to corresponding Traffic Detector. The Traffic Detector consists of four components: feature extractor extracts features from the network traffic, fuzzy rule base is the knowledge base using which fuzzy decision trees are built, rule base updater which adds new samples to the fuzzy decision tree that has been constructed and also adjusts the optimal tree size, and fuzzy inferencer fuzzifies the input values and processes them with the rule base. Forensic Analyzer includes collecting relative event data, analyzing correlated information relating with the event, and establishing digital evidences. IFDTNFS automated network forensic system which can produce interpretable and accurate results for forensic experts by applying a fuzzy logic-based decision tree data mining system.

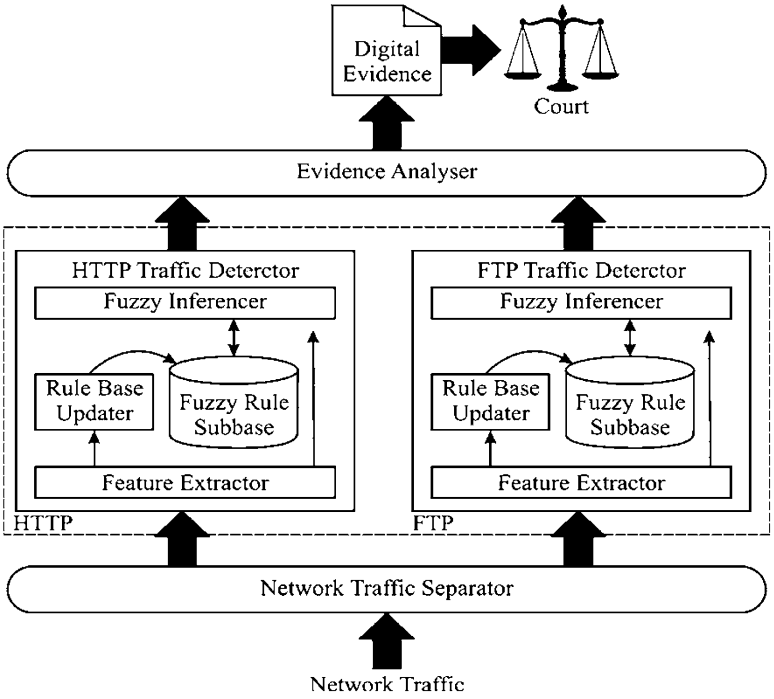

Fig. 3.5 Incremental fuzzy decision tree technology network forensic system

Zhang et al. [9] propose network forensic computing based on artificial neural network–principal component analysis (ANN-PCA). The major challenge faced in network forensics is massive information to be stored and analyzed. Extraction of key features reduces the storage by correlating the features with attacks. ANN-PCA techniques are used to identify all possible violations, extract features, and build signatures for new attacks. Classification is done using the FAAR algorithm to mine association rules and calculate the PCA values. Classification accuracy increases and information storage size decreases after feature extraction is performed using ANN-PCA.

Neuro-fuzzy techniques were used by Anaya et al. [10], to address the challenges of enormous data to be logged and analyzed for network forensic computing. The neuro-fuzzy solution is based on artificial neural network (ANN) and fuzzy logic and is used for evidence differentiation into normal and abnormal flows. ANNs are used in information processing to learn from the data and later generalize a solution. Fuzzy logic is used to generate a grade of membership to different behaviors so that attacks are determined. The model consists of four modules. The monitor control module stores all the network information. Information preprocessing module is made up of syntax sub-module and correlation sub-module. Syntax module is responsible for normalizing the inputs, and correlation sub-module aggregates the

different flow formats and groups the PDUs into a flow. Dependencies module relates all network element logs and takes decision about the flows. The decider module distinguishes between normal and abnormal flows. Recurrent neural network (RNN) was used to decide the type of flow.

Liao et al. [11] propose a network forensic system-based fuzzy logic and expert system (NFS-FLES), an effective and automated analyzing system which guarantees evidence reliability by collecting information from different sensors. It also analyzes computer crimes and makes automatic digital evidence using the approach of fuzzy logic and expert systems. The NFS-FLES consists of the following components – traffic capture, feature extractor, fuzzification, fuzzy inference engine, knowledge base, defuzzification, and Forensic Analyzer. The whole operation is done in four parts – real-time forensic data acquisition and preprocessing, knowledge base construction and dynamic rule generation, fuzzy linguistic operation of input attack data and computing aggregation fuzzy value, and total fuzzy score of every kind of attack. The forensic result is then output in time. A more efficient method for anomaly intrusion detection and real-time network forensics is to be researched. A multi-criteria forensic expert system which can build global and accurate classifier is to be developed. This forensic system is shown in Fig. 3.6.

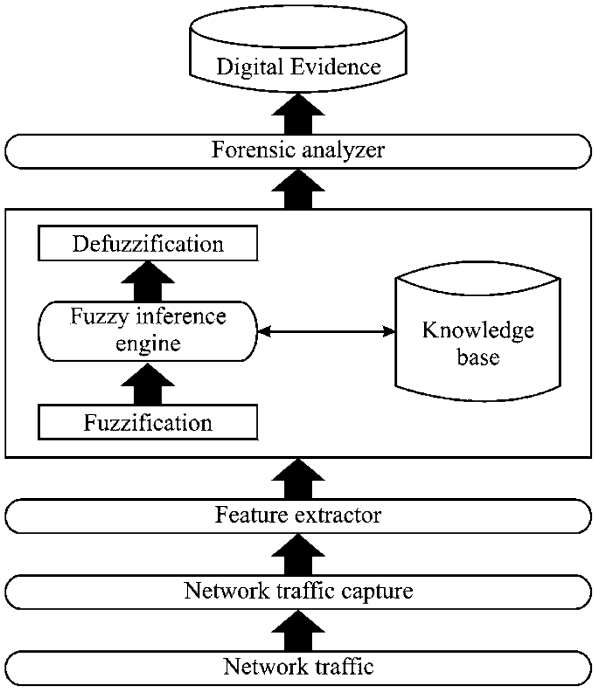

Fig. 3.6 The architecture of the proposed network forensic system

3.3 Honeynet-Based Frameworks

The honeynet system tries to find an unknown attack through an organized way in controlled system. It collects information from intruders. This section describes honeynet-based frameworks which are designed for network forensics. Honeytraps as a deception tool and honeynet data are described as well.

Honeytraps were proposed by Yasinsac and Manzano [12] as a deception tool to collect information about black hat activities and learn their techniques so that protection and defense mechanisms can be formulated. The production system forensic investigation system is represented in Fig. 3.7. Honeytraps are honeypot or honeynet systems which attract intruders to enter the host by emulating a known vulnerability. Once an attacker penetrates a honeytrap, data is captured to detect and record his actions. This data can be used to profile the tools and tactics used by the attackers putting the investigators into an offensive mode. Two architectures, serial and parallel, facilitate the forensic investigation. The serial architecture places the honeytrap between the Internet and the production system. Recognized users are filtered to the production systems, and the black hats are contained in the honeytrap. The parallel architecture allows the honeytrap to be independent of the production system. Once the system detects the presence of black hat, the

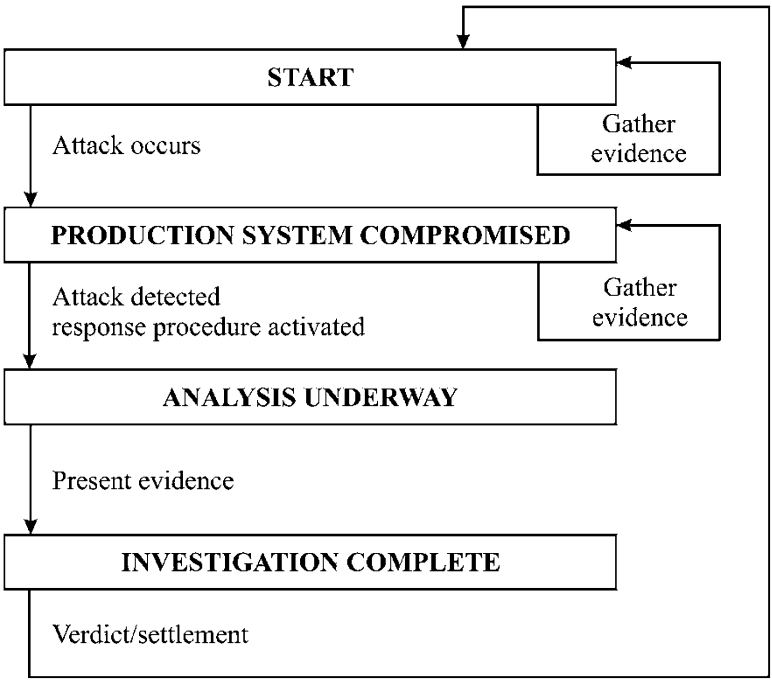

Fig. 3.7 Production system forensic investigation

forensic alert system is activated. If the attack is detected, forensic processes are activated on the honeytrap and production systems. Once the attack is contained, the investigation process is begun to determine the identity of the intruder on the production system.

Thonnard and Dacier [13] proposed a framework for attack patterns' discovery in honeynet data. Their work aims at finding groups of network traces sharing various kinds of highly similar patterns within an attack data set. They design a flexible clustering tool and analyze one specific aspect of the honeynet data, the time series of the attacks. Malicious network traffic is obtained from the distributed set of honeynet responders. Time signature is used as a primary clustering feature, and attack patterns are discovered using attack trace similarity. Attacks are detected as a series of connections, zero-day and polymorphic attacks are detected based on similarity to other attacks, and knowledge from the honeynet data can be leveraged in intrusion detection efforts. The clustering method does feature selection and extraction, define a pattern proximity measure, and group similar patterns. The result of the clustering applied to time series analysis enables detection of worms and botnets in the traffic collected by honeypots.

3.4 Attack Graph-Based Frameworks

Graph-based technology is also used to present network forensic framework. In this section, a novel graph-based approach for forensic analysis is discussed.

Wang et al. [14] develop a novel graph-based approach toward network forensic analysis. An evidence graph model facilitates evidence presentation and automated reasoning. The basic architecture has six modules: *evidence collection module* collecting digital evidence from heterogeneous sensors deployed, *evidence preprocessing module* transforms evidence into standardized format, *attack knowledge base* provides knowledge of known exploits, *assets knowledge base* provides knowledge of the networks and hosts under investigation, *evidence graph manipulation module* generates the evidence graph, and *attack reasoning module* performs semiautomated reasoning based on the evidence graph. A hierarchical reasoning framework consisting of two levels – local reasoning (functional analysis) aims to infer the functional states of network entities from local observations and global reasoning (structural analysis) aims to identify important entities from the graph structure and extract groups of densely correlated participants in the attack scenario. The results from both levels are combined and attacks further analyzed.

3.5 Formal Method-Based Frameworks

In this section, formal approaches are described for network forensic framework. Many technologies such as response probabilistic cognitive method, automated file fingerprinting, plug-in technique, dynamic forensic method, column-oriented

storage method, payload attribution method, carving technique, feedback mechanism, support vector regression, and self-organizing maps method.

Rekhis et al. [15] develop a system for digital forensics in networking (DigForNet) which is useful to analyze security incidents and explain the steps taken by the attackers. DigForNet uses the expertise of intrusion response teams and formal reasoning tools to reconstruct potential attack scenarios. They integrate the analysis performed by the IRT on a compromised system through the use of the Incident Response Probabilistic Cognitive Maps (IRPCMs). They also provide a formal approach to identify potential attack scenarios using investigation-based temporal logic of actions (I-TLA). They generate executable potential attack scenarios and show the progress of the attack using investigation-based temporal logic model checker (I-TLC), an automatic verification tool. Unknown attacks are handled by generating hypothetical actions. The generated executable potential attack scenarios are used to identify the risk scenarios that have compromised the system, entities which originated the attacks, different steps taken to conduct the attacks, and confirm the investigation. Methodology of proposed system is given in Fig. 3.8.

Haggerty et al. [16] have presented a method for automated file fingerprinting of malicious pictures resident on Web servers, which can be used for forensic purposes to check malicious digital pictures. This project is named as FORWEB, in which main components are fingerprint, fingerprint search, and fingerprint match. Web spider technology is used in FORWEB architecture to detect malicious actions. Fingerprint match function generates a report and run for malicious data with other information to investigate. Data are matched block by block and compared with fingerprints. A case study is also shown to check FORWEB where fingerprints of digital images are generated and searched over Web sites. Search time graph are shown for two Web sites; Flickr and ACSF sites. The model of FORWEB application is shown in Fig. 3.9.

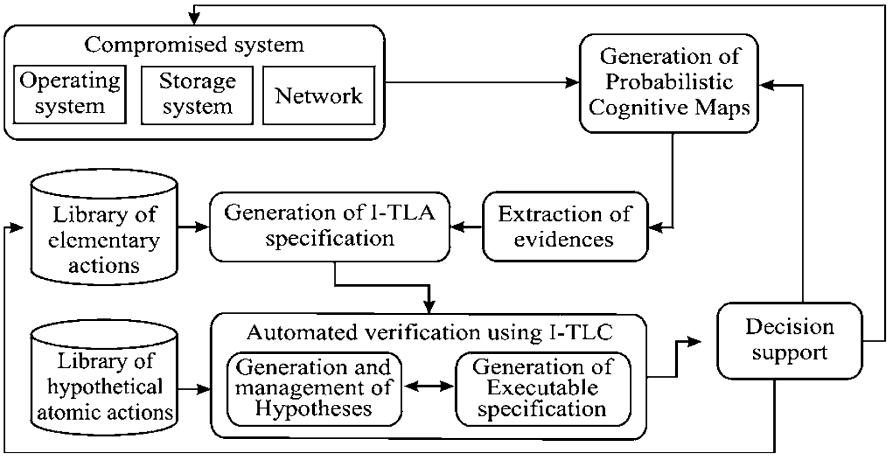

Fig. 3.8 DigForNet methodology

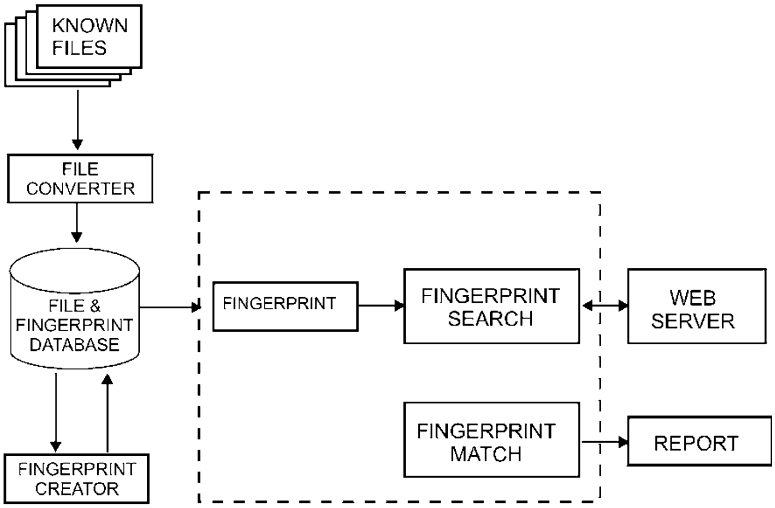

Fig. 3.9 Overview of the FORWEB application

Chen et al. [17] have presented a model to improve dynamic forensics with intrusion tolerance which can be able to find real-time evidences. This model is represented in Fig. 3.10. The components of the presented model work as when traffic comes to firewall, it redirects traffic to trigger (intrusion detection), actual server, and shadow server. If any malicious activity is found, evidence are collected and analyzed, and finally stored as evidence data in encrypted form. Formal description of intrusion detection system, shadow server, redirector, and evidence collector and analyzer is given in mathematical form. A case study over SITE EXEC vulnerability intrusion is performed, which results into improvement of actual server availability and forensic capabilities.

Giura et al. [18] have proposed a model named as NetStore for column-oriented storage infrastructure for network flow records, which does not need RDBMS specification, and it helps to reduce query processing time. NetStore has two components, which are processing engine and network flow column store. NetStore has three phases, buffering, segmenting, and query processing. NetStore also keeps record of internal IP index. Compression techniques such as run-length encoding, variable byte encoding, dictionary encoding, and frame of reference are used to reduce the size of each segments of NetStore. In query processing segment, NetStore supports only analytical queries and also responsible for network forensic and monitoring queries; it does not support transaction processing queries. Data insertion and query execution are the function of query processing segment. An evaluation of NetStore is presented; a comparison is also shown with PostgreSQL and LucidDB, where NetStore performs better than these tools.

Ponec et al. [19] proposed new methods for payload attribution for utilization of Rabin fingerprinting, shingling, and winnowing. Methods for payload attribution are

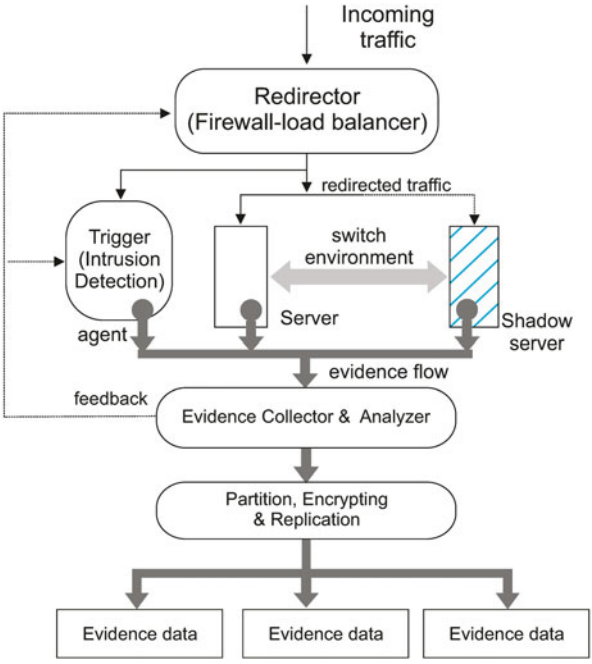

Fig. 3.10 Model for dynamic forensic-based intrusion tolerance

explained in detail, and these points are hierarchical bloom filter (HBF), fixed block shingling (FBS), variable block shingling (VBS), enhanced variable block shingling (EVBS), winnowing block shingling (WBS), variable hierarchical bloom filter (VHBF), variable doubles (VD), enhanced variable doubles (EVD), multi-hashing (MH), enhanced multi-hashing (EMH), and winnowing multi-hashing (WMH). Attacks on payload attribution system are explained as in terms of compression and encryption, fragmentation, boundary selection hacks, hash collisions, stuffing, resource exhaustion, and spoofing. Multi-packet queries, privacy and simple access control, and compression are also explained. Experimental results are also performed over a network trace of 4 GB of HTTP traffic data.

Hong et al. [20] have presented a framework for network forensics to seize and store the digital evidence of the escaped information in the network. The idea behind the framework is to collect the data, compress it, and then address the illegitimate information. Data are collected from fixed host and mobile host through agents and then it's compressed and restored in forensic center for forensic purpose. The process of forensic center is also discussed, which store data and analyze data. With a test of multimedia data, application data, text, and compressed data, it is shown that multimedia data are much higher than application data and text data.

Beverly et al. [21] have presented carving technique for IP packets and network data structure for forensic purpose and malware analysis. In this technique, it is suggested to create ground truth data, which is experimented with various operating systems. Carving signatures are developed with effectiveness measure of

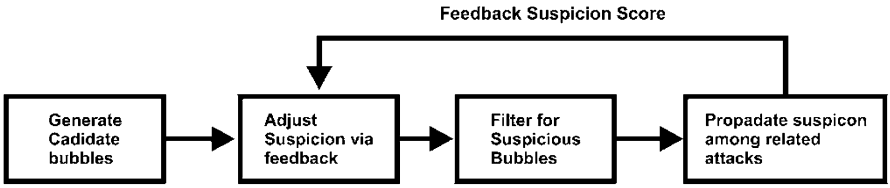

Fig. 3.11 Attack pattern discovery process

information retrieval system such as precision and recall. IP packets, socket structure, windows, and Ethernet are scanned through a new module for bulk_extractor (an open-source forensic tool), which is proved worthy. Filtering techniques, frequency analysis, correlation between modalities, and checksum are used to validate IP addresses and to manage false positives. Windows hibernation files are also decompressed through per-fragment decompression algorithm which improves IP address carving recall. In experimental results, a comparison against volatility and ground truth data is shown as an improvement in presented technique.

Zhu [22] has proposed a method to determine attack patterns using feedback mechanism. Forensic mining of network logs are performed after attack had launched. Proposed method is explained in four steps, at first generating candidate bubbles, adjusting suspicion via feedback, filtering for suspicion bubbles, and propagating suspicion among related attacks. Simulation and experiments are also explained in HTTP DoS attack environment to evaluate the performance of proposed method. Algorithm performs better accuracy while finding out attack pattern. Discovery process is shown in Fig. 3.11.

Chen et al. [23] have presented a method to do forensics in cognitive radio networks and in single channel. Packet arrival time prediction is done using support vector regression where three challenges are solved which is online prediction, overall optimization, and data capture and channel scan. Only those packets are captured which may be useful for forensic purposes. To accelerate the learning process, two methods are used which are incremental learning process and dual-regression function to reduce retraining. Monitor scheduling algorithm is also presented based on prediction result, and protocol for data capture in cognitive radio networks is explained. Performance of the presented method for packet arrival time prediction, data capture in small number of channels and large number of channels is evaluated to check the accuracy of the algorithm, which performs better result. A discussion is given over dynamics of secondary users, geographical coverage issues, and application-dependent packet prediction.

Ning et al. [24] have developed an analytical framework to compute the transmission evidence availability in network. Evidence maintenance, hop-level transmission evidence, path-level retransmission, accounting for retransmission, and bit rate selection are discussed in framework in the condition that transmitter and receiver do not abandon packets. How to explicitly compute the transmission evidence availability is also discussed through channel model and collision model.

A Forensic Analyzer is discussed to compute off-line, packet losses, transmission evidence availability, and analysis of misbehavior. Evaluation of the system is done through validating and simulating the analytical framework.

Palomo et al. [25] have presented a growing hierarchical self-organizing maps (GHSOMs)-based method for visualization and analysis of forensic or network data. Self-organizing maps (SOMs) have been used for data visualization in data mining applications. GHSOM provides a flexible way to visualize high-dimensional data. GHSOM is a hierarchically layered artificial neural network consisting of several SOMs. The GHSOM is trained with some dummy input data without supervision. Once trained, the GHSOM can later be used for better understanding and categorization of the network forensic data.

Garfinkel et al. [26] present a study of differential analysis and then apply that work to multiple contexts, including the analysis of files on a computer's disk drive, the pattern of data sent across a network, and even reports from other forensic tools. A general strategy for differential analysis is described as feature metadata, change primitives, temporal inconsistencies, and reporting. General algorithms are proposed which are categorized for different purposes such as idifference for finding differences between two different images, rdifferences for finding differences between two different registry hives, bulk_diff program for comparing histograms, corpus_sync program for synchronization of files, and flowdiff for finding differences between two pcap files. At last, file system differencing case study is done.

Stealth attacks specially crafted to evade intrusion detection techniques may aggravate the security risks. In this paper, Chen et al. [27] discuss the problem and feasibility of back tracking the origin of a self-propagating stealth attack when given a network traffic trace for a sufficiently long period of time. They develop a data reduction method based on host contact activities to filter out attack-irrelevant data and only retain evidence relevant to potential attacks for postmortem investigation.

Scanlon et al. [28] have introduced a universal peer to peer network investigation framework which is faster and need less effort in nature. A comparison is shown between centralized and distributed peer to peer network architectures. Peer to peer network investigation types are discussed in detail which includes evidence collection, anatomy, wide area measurement, and botnet takeover. In a universal peer to peer network investigation framework, there are three modules, traffic collection, traffic analysis module connected to traffic pattern database, and regular P2P client emulation module – this system is connected to server and regular peer to peer activity. Advantages of the proposed architecture are compatibility, cost, automated identification, and speed. Through traffic communication, peer exchange, distributed hash table, and local peer discovery, proof of the proposed technique is explained as well.

Gebhardt et al. [29] have presented a model for network forensics in cloud computing and identify challenges for infrastructure as services. Five basic layers are described in network forensic architecture for cloud, which are data collection, aggregation, analysis, and reporting. These layers are managed by management layer. A prototype implementation is shown using OpenNebula and VMM. Result

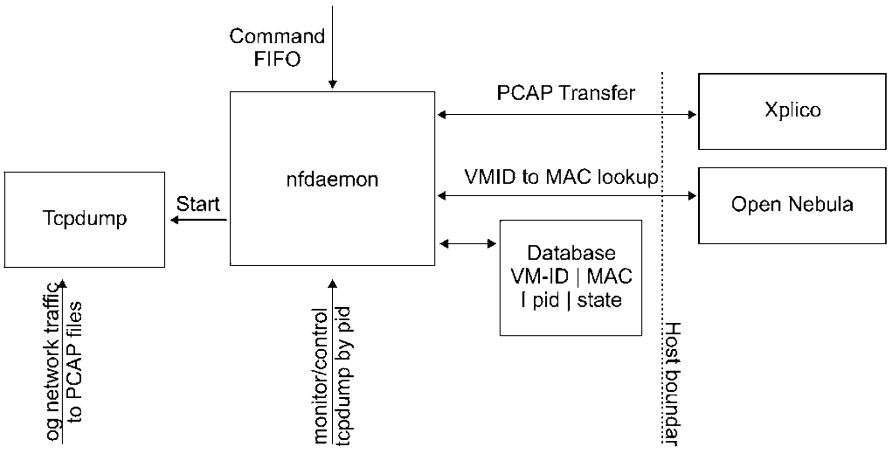

Fig. 3.12 Network forensic daemon and its components

is shown as proposed architecture provides facilities to cloud clients for network forensics remotely, it is able to work in multi-tenant environment, and it also reduces cost of transferring captured network data. The process of network forensic is shown in Fig. 3.12.

Huang et al. [30] have analyzed the static IP flow data to detect the malicious IP sources using unsupervised learning method named as growing hierarchical self-organizing map (GHSOM). Using this technique, attack patterns are analyzed, and attack behavior is also identified. Architecture and algorithm are shown for anomaly detection where data is passed through training and testing stage. Training stage includes five steps which are packets clustering signature labeling, feature extracting, IP visualization, and behavior visualization. Testing stage performs IP classification and behavior classification. To check the effectiveness and efficiency of the proposed work, data set from Cooperative Association for Internet Data Analysis (CAIDA) is considered. Some malicious events such as DoS attack, backscatter, Internet worms, and host scanning are added. Packet clustering, IP visualization, behavior visualization, IP filtering, and behavior filtering are performed, and it is shown that with use of GHOSM technique, network forensics can be performed in more efficient way. The model is shown in Fig. 3.13.

3.6 Aggregation-Based Frameworks

Aggregation-based approach is used to present network forensic framework. Many authors have presented network forensic framework based on aggregation approach. Data recorder at host level and network level, portable network forensic evidence collector, and packet analysis through Network Traffic Exploration are presented.

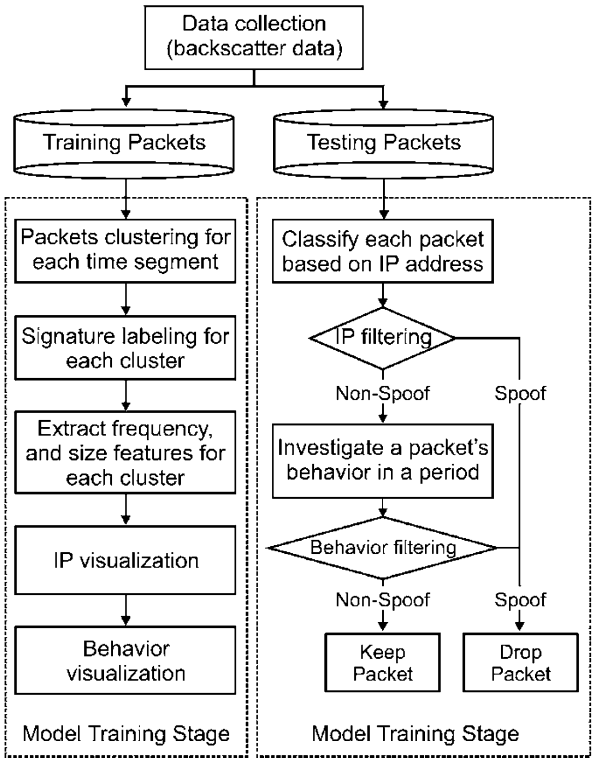

Fig. 3.13 Architecture of anomaly detection approach

Almulhem et al. [31] propose a network forensic system (NFS) that records data at the host level and network level. The system consists of three main modules – marking, capturing, and logging. Marking module decides whether a passing packet is malicious. One or more sensors (like IDS) report suspicious IP addresses. Capture module is a collection of lightweight capture modules which wait for the marked packets. They arrange to reliably transport them to the logging module for archival. Logging module is a systems repository where attack data are being stored. It uses three types of logger – hosts logger stores data sent by capture module, sensors logger stores sensors alerts, and raw logger is optional and is used when other loggers fail. The capture module was implemented using *Sebek*, marking module used *snort* IDS, and logging module used server-side *Sebek*; snort's *barnyard* tool was used to log the alerts to *MySQL* database, and *ACID Lab* was used for analysis. Finally *TCPDump* was used as raw logger. The alert module is still under development and will be implemented as an expert system. This will analyze logged data, assess, and reconstruct key steps of attacks. The process is shown in Fig. 3.14.

Nikkel [32] proposed a portable network forensic evidence collector (PNFEC) built using inexpensive embedded hardware and open-source software. The compact and portable device has been designed for traffic collection between a network and a single node, having specific modes of operation, rapid deployment, and stealthy

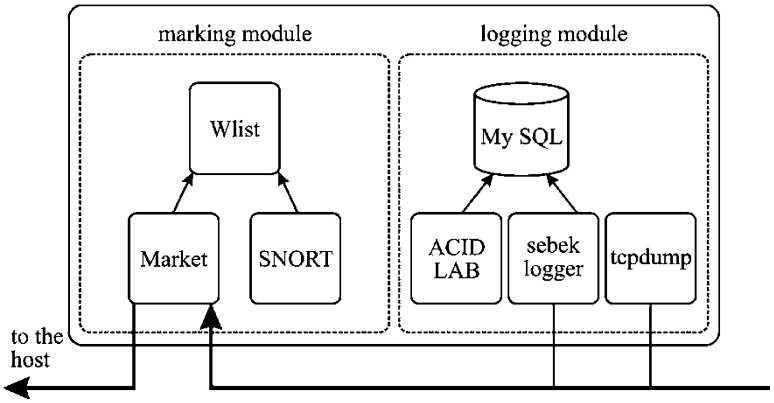

Fig. 3.14 The bridge internal

inline operation. The traffic on the Ethernet bridge is promiscuously captured using various pcap-based capture tools and stored on a hard disk. The operating system, additional software, configuration files, and investigator activity logs are stored on a compact flash. Administrative access controls various aspects of the device like start-up, scheduling, configuration of capturing filters, forensic functions such as preserving, and transferring the evidence. The PNFEC is easy to deploy and operate (plug and play). The network traffic collected can be stored in encrypted form. PNFEC also controls filtering of captured data using TCPDump to ensure there are no privacy violations. A script is used to create a cryptographic hash of the packet capture files and preserved. OpenBSD is used as the operating system; many of the functionalities like secure access, packet capture, encrypted file system, evidence preservation, disk wiping, and formatting tools are included by default. Tools for troubleshooting (TCPflow) and pcap management (TCPslice) are also added. PNFEC operates in three modes – investigator, server, and user. The device does not modify or inject traffic as it acts as a bridge at the link layer. Administrative interfaces are to be developed, the device may be available with other operating systems, data may be collected from other tools, and evidence disk capacity needs further development.

Vandenberghe [33] proposed a Network Traffic Exploration (NTE) application being developed by Defense Research and Development Canada (DRDC) for security event and packet analysis. This tool combines six key functional areas into a single package. It includes intrusion detection (signature and anomaly based), traffic analysis, scripting tools, packet playback, visualization features, and impact assessment. NTE has three layers with MATLAB as development environment, low-level packet analysis library, and unified application front end. It provides an environment where statistical analysis, session analysis, and protocol analysis can exchange data. The process is shown in Fig. 3.15.

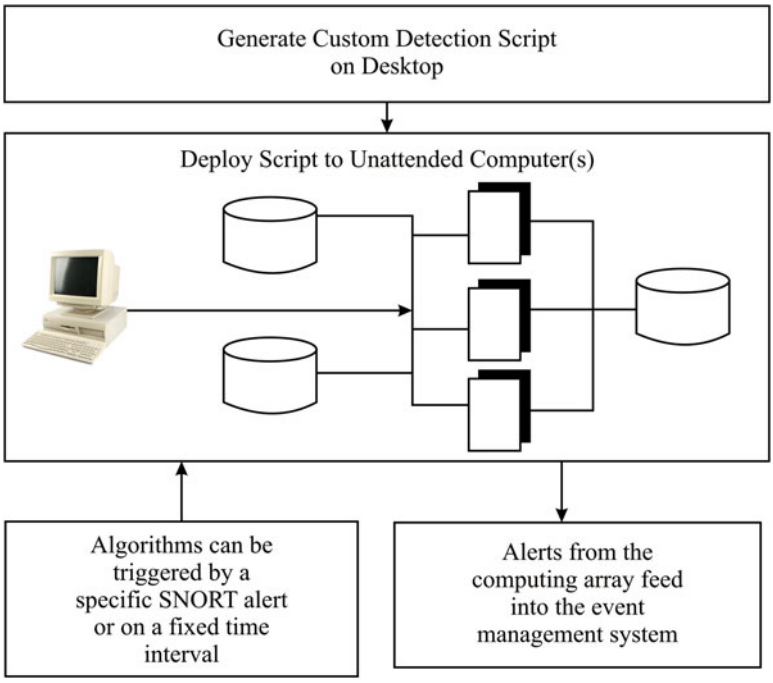

Fig. 3.15 Network Traffic Exploration Process workflow

3.7 Data Mining-Based Frameworks

Data mining algorithms are also used in network forensic framework. In this section, association rule mining is discussed which is used to detect malicious activity from network.

Brauckhoff et al. [34] have used association rule mining to detect anomalous activity from network. Histogram-based detectors are used to identify suspicious flows. NetFlow data set is used to evaluate the proposed technique, where classification cost is shown as it is reduced. The problem is tried to identifying the traffic flows associated with an anomaly. Histogram-based detection, metadata generation, and association rule mining with a priori algorithm and its use in presented model is explained. The forensic framework is explained as configuration file, and forensic signal is interacted to XML parse which goes into plug-in. This plug-in part also takes input from capture file. It then goes into encrypt signature and makes a library of raw evidence. With the use of association rules, classification cost can be minimized. The model is shown in Fig. 3.16.

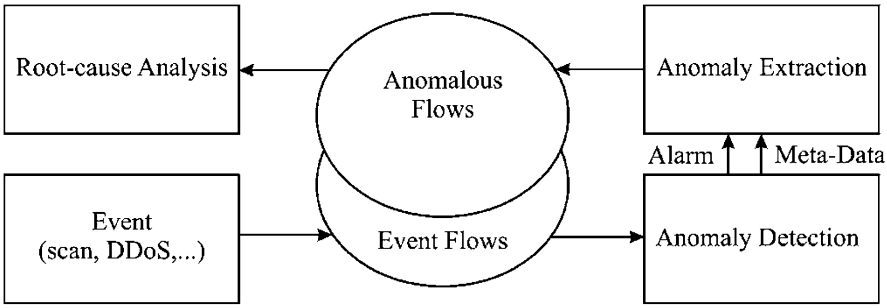

Fig. 3.16 High-level goal of anomaly extraction

3.8 Conclusion

In this chapter, network forensic frameworks were discussed which are actual implementations using different techniques. These techniques can be applied in hybrid approach using open source tools so that results can be produced more effectively. A new Network forensic system can be built to overcome research gaps and challenges in existing implementations. The framework can focus on the following phases of the generic process model – traffic collection; detection of attack features; data fusion of various attack attributes; examination of network traces; analysis using soft computing and data mining approaches; attack investigation and attribution. The implementation can be compared with existing proprietary tools and generic hardware can be added to make it a fully functional Network forensic analysis tool.

3.9 Questions

Multiple Choice Questions

Select the most suitable answer for the following questions:

1. ForNet can identify network events like ____

   (a) ICMP messages
   (b) TCP connection establishment, port scanning
   (c) UDP connection establishment
   (d) Both TCP and UDP

2. PNFEC stands for ____

   (a) Portable network forensic evidence collector
   (b) Partial network forensic evidence connector
   (c) Portable network forensic evidence connector
   (d) Partial network forensic evidence collector

3. OpenBSD is ____

   (a) Operating system
   (b) Programming language
   (c) Antivirus
   (d) Firewall

4. CAIDA stands for ____

   (a) Cooperative association for Internet data analysis
   (b) Cooperative activity for Internet data analysis
   (c) Cooperative association for Internet data assignment
   (d) Cooperative activity for Internet data assignment

5. Network forensic agents are engines of ____

   (a) Data gathering
   (b) Data analysis
   (c) Help in anti-forensics
   (d) Attribution

6. Marking module performs ____

   (a) Identification of outgoing packets
   (b) Rejection of automatic events
   (c) Identification of malicious activity
   (d) Analysis of evidence

7. Bulk_extractor is ____

   (a) Forensic tool
   (b) Security tool
   (c) Antivirus
   (d) Firewall

8. PNFEC operates in ____

   (a) Two modes
   (b) Three modes
   (c) Four modes
   (d) Five modes

9. GHSOM stands for ____

   (a) Growing hierarchical self-organizing method
   (b) Growing hierarchical self-organizing market
   (c) Growing hash value self-organizing map
   (d) Growing hierarchical self-organizing map

10. Forensic mining of network logs are performed ____

    (a) Parallel to the system
    (b) In real-time case

(c) After attack had launched
(d) Before attack had launched

Short-Answer Questions

1. Write the brief answers of following questions:
2. Write in detail answer of following questions:
3. Briefly describe the distributed systems-based frameworks.
4. What is distributed agent? Give the model of distributed cooperative network forensic system.
5. What is dynamical network forensics? How is it implemented? What is the role of forensic agent in it?
6. Compare the soft computing-based techniques implemented in network forensic framework.
7. What are the components of network forensic system-based fuzzy logic and expert system? Explain each of them.

Long-Answer Questions

1. What is Incident Response Probabilistic Cognitive Maps?
2. How to use intrusion tolerance to find real-time evidences? Give the description of intrusion detection system.
3. Compare the different singling methods. How can it help to network forensics?
4. What is stealth attack? What techniques are used to avoid this attack?
5. Compare the centralized and distributed peer to peer network architectures.
6. Give the architectural view of portable network forensic evidence collector (PNFEC).

References

1. Shanmugasundaram K, Memon N, Savant A, Bronnimann H (2003) ForNet: a distributed forensics network. In: Gorodetsky V, Popyack L, Skormin V (eds) Computer network security, vol 2776. Springer, Berlin/Heidelberg, pp 1–16
2. Ren W (2004) On a network forensics model for information security. In: 3rd international conference on Information Systems Technology and its Applications (ISTA 2004), Utah, USA, pp 229–234
3. Jing YN, Tu P, Wang XP, Zhang GD (2005) Distributed-log-based scheme for IP traceback. In: The fifth international conference on Computer and Information Technology (CIT' 05), Shanghai, China, pp 711–715
4. Tang Y, Daniels TE (2005) A simple framework for distributed forensics. In: 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS 05), Columbus, OH, USA, pp 163–169
5. Nagesh A (2006) Distributed Network Forensics using JADE Mobile Agent Framework. M. S. thesis, Dept of Computing Studies, Arizona State University, Mesa, AZ
6. Wang D, Li T, Liu S, Zhang J, Liu C (2007) Dynamical network forensics based on immune agent. In: Third International Conference on Natural Computation (ICNC 2007), Haikou, Hainan, China, pp 651–656

7. Kim JS, Kim M, Noh BN (2004) A fuzzy expert system for network forensics. In: Laganà A, Gavrilova ML, Kumar V, Mun Y, Tan CJK, Gervasi O (eds) Computational science and its applications, vol 3043. Springer, Berlin/Heidelberg, pp 175–182
8. Liu Z, Feng D (2005) Incremental fuzzy decision tree-based network forensic system. In: Hao Y, Liu J, Wang YP, Cheung YM, Yin H, Jiao L, Ma J, Jiao YC (eds) Computational intelligence and security, vol 3802. Springer, Berlin/Heidelberg, pp 995–1002
9. Zhang Y, Ren Y, Wang J, Fang L (2007) Network forensic computing based on ANN-PCA. In: International conference on Computational Intelligence and Security Workshops (CISW 07), Harbin, Heilongjiang, China, pp 942–945
10. Anaya EA, Nakano-Miyatake M, Perez Meana HM (2009) Network forensics with Neurofuzzy techniques. In: 52nd IEEE international Midwest Symposium on Circuits and Systems (MWSCAS '09), Cancun, Mexico, pp 848–852
11. Liao N, Tian S, Wang T (2009) Network forensics based on fuzzy logic and expert system. Comput Commun 32(17):1881–1892
12. Yasinsac A, Manzano Y (2002) Honeytraps, a network forensic tool. In: 6th world multiconference on Systemics, Cybernetics, and Informatics (SCI 02), Florida, USA
13. Thonnard O, Dacier M (2008) A framework for attack patterns' discovery in honeynet data. Digit Investig 5(Supplement 1):S128–S139
14. Wang W, Daniels TE (2008) A graph based approach toward network forensics analysis. ACM Trans Inf Syst Secur (TISSEC) 12(1):4
15. Rekhis S, Krichene J, Boudriga N (2008) DigForNet: digital forensic in networking. In: Jajodia S, Samarati P, Cimato S (eds) IFIP TC-11 23rd international information security conference, vol 278. Springer, Boston, pp 637–651
16. John H, David LJ, Mark T (2008) FORWEB: file fingerprinting for automated network forensics investigations. In: 1st international conference on forensic applications and techniques in telecommunications, information, and multimedia and workshop, Adelaide, Australia
17. Lin C, Zhitang L, Cuixia G, Yingshu L (2009) Modeling and analyzing dynamic forensics system based on intrusion tolerance. In: Ninth IEEE international conference on computer and information technology, pp 230–235
18. Jha S, Sommer R, Kreibich C, Giura P, Memon N (2010) NetStore: an efficient storage infrastructure for network forensics and monitoring. In: Recent advances in intrusion detection, vol 6307. Springer, Berlin/Heidelberg, pp 277–296
19. Miroslav P, Paul G, Joel W, Herv B (2010) New payload attribution methods for network forensic investigations. ACM Trans Inf Syst Secur 13(2):1–32
20. Tang H, Zou T, Jin Q, Zhang J (2011) A distributed framework for forensics based on the content of network transmission. In: First international conference on instrumentation, measurement, computer, communication and control, pp 852–855
21. Beverly R, Garfinkel S, Cardwell G (2011) Forensic carving of network packets and associated data structures. Digit Investig 8(Supplement):S78–S89
22. Ying Z (2011) Attack pattern discovery in forensic investigation of network attacks. IEEE J Sel Areas Commun 29(7):1349–1357
23. Chen S, Zeng K, Mohapatra P (2011) Efficient data capturing for network forensics in cognitive radio networks. IEEE/ACM Trans Netw PP(99):1–1
24. Jianxia N, Singh S, Pelechrinis K, Liu B, Krishnamurthy SV, Govindan R (2012) Forensic analysis of packet losses in wireless networks. In: 20th IEEE international conference on network protocols, pp 1–10
25. Palomo EJ, Elizondo D, Dom'ınguez E, Luque RM, Watson T (2012) SOM-based techniques towards hierarchical visualisation of network forensics traffic data. In: Computational intelligence for privacy and security, vol 394. Springer, Berlin/Heidelberg, pp 75–95
26. Garfinkel S, Nelson AJ, Young J (2012) A general strategy for differential forensic analysis. Digit Investig 9(Supplement, no. 0):S50–S59
27. Chen LM, Chen MC, Liao W, Sun YS (2013) A scalable network forensics mechanism for stealthy self-propagating attacks. Comput Commun 36(13):1471–1484

28. Scanlon M, Kechadi MT (2013) Universal peer-to-peer network investigation framework. In: Eighth international conference on availability, reliability and security, pp 694–700
29. Gebhardt T, Reiser HP (2013) Network Forensics for Cloud Computing. In: 13th IFIP WG 6.1 international conference on Distributed Applications and Interoperable Systems, Florence, Italy, pp 29–42
30. Shin-Ying H, Yennun H (2013) Network Forensic Analysis Using Growing Hierarchical SOM. In: 13th international conference on Data Mining Workshops, pp 536–543
31. Almulhem A, Traore I (2005) Experience with engineering a network forensics system. In: International conference on Information Networking, Convergence in Broadband and Mobile Networking (ICOIN 05), Jeju Island, Korea, pp 62–71
32. Nikkel BJ (2006) A portable network forensic evidence collector. Digit Investig 3(3):127–135
33. Vandenberghe G (2008) Network traffic exploration application: A tool to assess, visualize, and analyze network security events. In: Goodall J, Conti G, Ma K-L (eds) Visualization for computer security, vol 5210. Springer, Berlin/Heidelberg, pp 181–196
34. Brauckhoff D, Dimitropoulos X, Wagner A, Salamatian K (2009) Anomaly extraction in backbone networks using association rules. In: Internet Measurement Conference (IMC), pp 1788–1799