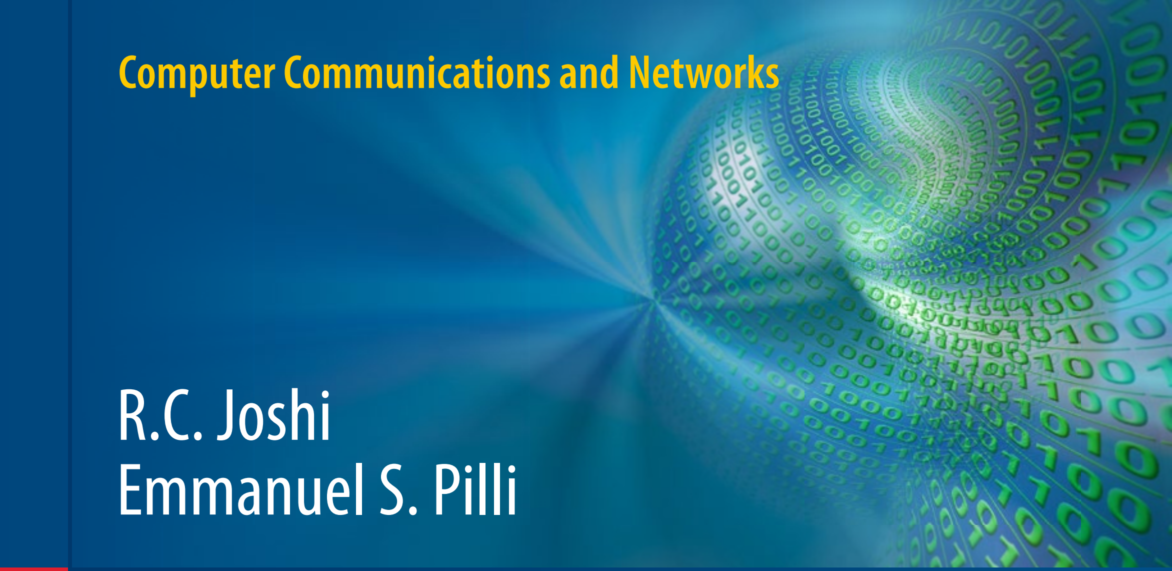
Computer Communications and Networks



R.C. Joshi
Emmanuel S. Pilli

Fundamentals of Network Forensics

A Research Perspective

Springer

Computer Communications and Networks

Series editor
A.J. Sammes
Centre for Forensic Computing
Cranfield University, Shrivenham Campus
Swindon, UK

The **Computer Communications and Networks** series is a range of textbooks, monographs and handbooks. It sets out to provide students, researchers, and non-specialists alike with a sure grounding in current knowledge, together with comprehensible access to the latest developments in computer communications and networking.

Emphasis is placed on clear and explanatory styles that support a tutorial approach, so that even the most complex of topics is presented in a lucid and intelligible manner.

More information about this series at http://www.springer.com/series/4198

R.C. Joshi • Emmanuel S. Pilli

Fundamentals of Network Forensics

A Research Perspective

Springer

R.C. Joshi
Graphic Era University
Dehradun, Uttarakhand, India

Emmanuel S. Pilli
Malaviya National Institute of Technology
Jaipur, Rajasthan, India

ISSN 1617-7975 ISSN 2197-8433 (electronic)
Computer Communications and Networks
ISBN 978-1-4471-7297-0 ISBN 978-1-4471-7299-4 (eBook)
DOI 10.1007/978-1-4471-7299-4

Library of Congress Control Number: 2016951437

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer-Verlag London Ltd.
The registered company address is: 236 Gray's Inn Road, London WC1X 8HB, United Kingdom

Dedicated to

To my brother Late Shri A. B. Joshi

—R. C. J

To my father Late Prof. Pilli Alfred James and my mother Late Mrs. Pitta Cecelia Nancy James

—E. S. P

and all faculty and students who are working hard to make the cyberworld a safe place

Preface

Introduction

Network security is rapidly becoming mainstream activity in an increasingly online society due to ubiquity of computer networks. We use computers, laptops, and smartphones daily and communicate over the Internet to access various applications. We connect to web servers, e-mail servers, and network servers to access various applications which may be for our personal use. These accesses over the Internet creates a sand of data. The Internet has long since passed the point where we can fully analyze and comprehend its working. We can understand bits and pieces of it and can make broad generalization, but the fact is that we humans have already created a far more powerful and complex system. In this environment, a new field of study has evolved: network forensics. Network forensics generally refers to the scientific study of network-based evidence. Network forensics is a field of study independent of any special legal case and understanding many of the scientific advances, tools, and techniques developed for the purposes of legal investigation.

Network forensics shows how to find the clues behind an Internet crime scene. We can learn how to uncover information that lies hidden in every e-mail message, web page, and web server over the Internet. A growing interest in this field has been motivated by many factors. Cyberattacks like viruses and worms, compliance of many regulations, and assuring common man of security are some of them. By continuous analysis of many similar problems occurring, we can understand how the Internet and its core protocols are being exploited maliciously. We can identify where shortcomings lie and where hardening has to be made.

In this book, we have endeavored to provide a technical foundation that will be practically useful not just for professional network forensics analysts conducting legal investigations but also for students, independent researchers, and all those who are curious.

Audience

This book is intended for both academic and professional audience. As a textbook, it is intended as a semester course at graduate level in Computer Science, Information Technology, Network Security, and Information Science and Management. The book serves as basic reference volume for researchers in network forensics. It will be useful to practitioners, forensic investigators, and the incident response teams.

To get the most out of this book, the reader should have a working knowledge of various operating system environments, programming languages Perl, Python, and Java and a working knowledge of security tools.

Organization of the Book

The book is organized to provide a broad overview of the important topics of network forensics. It is divided into three parts: "Fundamentals," "Techniques," and "Advances."

Part I, "Fundamentals," covers the basic concept of network process models, network forensics frameworks, and network forensics tools. This provides a foundation for more advanced topics, which are covered in the next two parts. Part I includes the following chapters:

Chapter 1: "Network Forensics" presents an introduction to the book and also to the nascent discipline. The chapter also discusses the difference between network security and network forensics.

Chapter 2: "Network Forensic Process Models" presents various process models along with their various phases in the field of digital forensics and network forensics.

Chapter 3: "Network Forensic Framework" presents various frameworks based on various criteria and their relevance with the network forensics.

Chapter 4: "Network Forensic Tools" presents various tools that are available in the literature for doing the network forensics.

Part II, "Techniques," discusses the major mechanisms which can be applied to the network forensics.

Chapter 5: "Network Forensic Acquisition" covers the Collection of network packets and traffic in any computer network or system.

Chapter 6: "Network Forensic Analysis" covers the activities required to perform analysis. Misuse detection and Anomaly detection techniques make up, most of the analysis techniques.

Chapter 7: "Network Forensic Attribution" discusses various traceback techniques required for attributing an attack to a source.

Part III, "Advances," covers various advance topics such as botnet forensics, smartphone forensics, and cloud forensics.

Chapter 8: "Botnet Forensics" covers the botnet threat, architectures, protocols, and lifecycle of botnet investigation of attacks based on botnets is still a challenge are discussed.

Chapter 9: "Smartphone Forensics" covers the standard process model which consists of preservation, acquisition, examination, and analysis, and reporting of retrieved data is discussed. Various frameworks for smartphone forensics are also discussed. Smartphone forensic tools are also discussed. In the last section, the research challenges related to smartphone forensic are also discussed.

Chapter 10: "Cloud Forensics" covers cloud forensic challenges and research directions. A generic process model for cloud forensic is discussed along with four phases, namely, identification, collection, acquisition, and preservation.

Tools

This book is designed to be accessible to a wide audience to teach the fundamental principles and techniques of network forensics. There are many tools available to perform various forensic activities. The focus was to include the tools which are freely available and easy to setup. Chapter 4 gives various tools to perform various activities of network forensics.

Dehradun, Uttarakhand, India R.C. Joshi
Jaipur, Rajasthan, India Emmanuel S. Pilli

Acknowledgements

This book is a result of our research work at the Indian Institute of Technology Roorkee, Graphic Era University Dehradun and Malaviya National Institute of Technology Jaipur. The book has been inspired by our popular paper in the Journal of Digital Investigation (Elsevier) titled "Network forensic frameworks: Survey and research challenges". We would like to offer special thanks to our coauthor, Dr. Rajdeep Niyogi (Associate Professor, IIT Roorkee). We have thoroughly updated our work and also included three advanced topics on Botnets, Smartphones and Cloud.

We would like to thank Prof. Kamal Ghanshala, President, Graphic Era University – Dehradun, Prof. I. K. Bhat, former Director MNIT Jaipur and Prof. Udaykumar R Yaragatti, Director MNIT Jaipur who have wholeheartedly supported the writing of this book.

We would like to thank many research associates, research scholars, graduate & undergraduate students, and interns who assisted us in this work. In particular, we would like to thank Preeti Mishra, Anand Kumar Mishra, Anchit Bijalwan, Ramesh Rawat, Sameer Chauhan, Priya Agrawal, Nanak Chand, Dhyanendra Rathore, Avani Sharma, Tanvi Chawla, Madhuri Gupta, and Saurabh Dwivedi for their effort and hard work. They contributed a lot in literature survey, original wording and editing of what now has become this book. They have done most of the art work as well. We also thank Satyender Singh and Suraj Soni for assisting us in the office.

We gratefully recognize the opportunity Springer-Verlag gave us to write this book. We like to make a special mention of thanks to all the splendid staff at Springer-Verlag, London who put so much time and effort into producing this book, especially the team for the series on Computer Communications and Networks – Wayne Wheeler (Senior Editor) and Simon Rees (Associate Editor). The patience and forbearance shown by Simon Rees, when schedules were difficult to meet, needs special appreciation. We want to thank Wayne and Simon for believing in us. We would also want to thank Tirupthirekha Das Mahapatra (Production Editor) for overseeing the publication of the complete manuscript.

A special mention of thanks goes to the excellent team at SPiGlobal for producing this book, especially Atma Biswal (Project Manager Publishing) and

Albert André (Account Manager) for their patience and kindness. They were ever ready to incorporate many editing changes made by us during the proof reading phase. We are very grateful to our friends and colleagues who conducted technical reviews of the book.

Prof. Joshi would like to thank all his family members especially his wife Smt. Usha Joshi and daughters Ira and Bakul.

Dr. Emmanuel Pilli would like to thank his family members (Renuka and Julius, Kamala and Daniel, Martha and Samuel, Debbie and Sunil, Ashwathy and Ebenezer) and colleagues in MNIT and GEU (Prof. M. C. Govil, Rajarshi Mahapatra, Dibyahash Bordoloi, Ravi Gowri, Rishi Prakash, Girish Lakhera, Pravin Patil, Dinesh Tyagi, Santosh Vipparthi, and Arka Mazumdar) for their constant motivation and reminder to complete the book. Special thanks to many friends (Havilah and Ravikalyan, Pallavi and Vimal, Jessy and Jacob, Mercy and Deepak Nelson, Grace and Sunder, Rajini and Srinivas, Geetha and Arun, Nisha and Alex, Devendra, Samuel Joshua, Suresh, Pheobe, Preethi, Phebe, Swathi, Ashish, Rosaiah, Ravikanth, Shalini, Gracy, Jessi, and Surendra) who supported to push the book to completion. Lastly very special thanks to the two most important people of his life, wife Phoebe Vanmathy Julius and their daughter Pramiti Evangeline for the great sacrifice which they made, of the time he spends with them, while this book was being written, edited, and proof read.

R. C. Joshi
Emmanuel S. Pilli

Contents

Part I Fundamentals

Part II Techniques

Part III Advances

Part I
Fundamentals

Chapter 1
Network Forensics

Learning Objectives

- Overview of the standard terms in network forensics
- Understand the background, definition, and classification
- Understand the motivation and applications
- Explore the emerging and challenging areas of research

1.1 Introduction

On November 24, 2014, *Los Angeles Times* reported that employees of the Culver City-based studio of Sony Pictures Entertainment were getting a message with threatening warning when they tried to log on their work computers. An image of a sneering red skeleton appeared on the screen under “Hacked By #GOP,” reportedly short for “Guardians of Peace” [1]. Sony Pictures Entertainment was hacked, and employees’ personal information and information about their dependents, email communication between employees, executive salary information, and copies of unreleased Sony films among other information were released.

FBI Director James Comey said a group of shadowy hackers, Guardians of Peace, blamed by the United States for the computer attack against Sony Pictures Entertainment “got sloppy” and left behind clues that point to North Korea’s involvement. The group had previously sent threatening e-mails to Sony, using an Internet provider address used exclusively by North Korea. Comey said that security experts who were not agreeing to his view don’t have the facts that he had and don’t see what he saw.

However various security experts examined the evidence left behind by the attackers, and their research provided insight into the source of these attacks. Though not definitive, their analysis provided a much clearer picture and suggested an organized criminal group operating out of Romania responsible for the data

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_1

breach impacting Sony Pictures Entertainment. The experts were able to reconstruct the attack from the ground up and discovered a number of IP addresses that were linked to other attacks that have been attributed to actors in Romania as well. The presence of Romanian text in the comment strings of the malware was recovered during the forensic investigation [2].

They found that malware was delivered using a “spear phishing” message targeted at top-level executives on November 13, 2014. A day later, the malware began communicating with C2 server and began spreading by means of SMB shares and identifications gained from the C2. Nine days later, an account called GuardiansOfPeace logged into pastebin. Initial release of confidential data took place a day later, on November 2014.

Security expert Bruce Schneier who runs a successful security blog asked the following questions about the incident after analyzing that FBI was confident because of NSA trying to eavesdrop on North Korea’s government communications: Should the National Security Agency defend US corporate networks or US military networks? How much should organizations like the NSA be allowed to insist that they be trusted without proof when they claim to have classified evidence that can’t be disclosed? More importantly, when it is not known who is launching an attack and why [3]?

Similar reports surfaced in February and May 2013 against China’s People’s Liberation Army Unit 61398 for an overwhelming percentage of the attacks on American corporations, organizations, and government agencies. American intelligence officials confirmed a growing body of digital forensic evidence in that small locality [4, 5].

On April 25, 2011, Iran has been targeted by a new computer worm named “Stars,” which is companionable with the beleaguered system, causes minimal harm in the initial stage, and the worm is likely to be mistaken for executable files of the government [6]. “Stars” is the second computer worm to target Iran after the “Stuxnet” worm, which was capable of taking over power plants and had infected many industrial sites. W32.Stuxnet worm has been in the focus of media and researchers. Stuxnet was discovered in June/July 2010 and is one of the complex threats in recent times. It targets industrial control systems and modifies code on programmable logic controllers (PLCs) to make them work in a manner the attacker intends to [7].

Stuxnet utilized antivirus evasion techniques, complex process injection code, four separate zero-day vulnerabilities, and the first ever rootkit designed specifically for PLC systems [8]. It spread via unpatched holes in Windows and USB devices, dropped the rootkit to hide the compromise from administrators, and used fraudulent digital certificates to pose as trusted software [9]. *Stuxnet* targeted PLCs on sites using Siemens SIMATIC WinCC or STEP 7 Supervisory Control and Data Acquisition Systems (SCADA). The *Stuxnet* computer worm might have been designed specifically to attack Iran’s nuclear program as it infiltrated industrial systems mostly in Iran and potentially crippled centrifuges used to enrich uranium [10]. Figure 1.1 shows number of users infected with the Rootkit.Win32.Stuxnet.

Stuxnet exploited numerous Windows vulnerabilities and at least four of them are zero-day vulnerabilities (MS08-067 RPC Exploit, MS10-046 LNK Exploit,

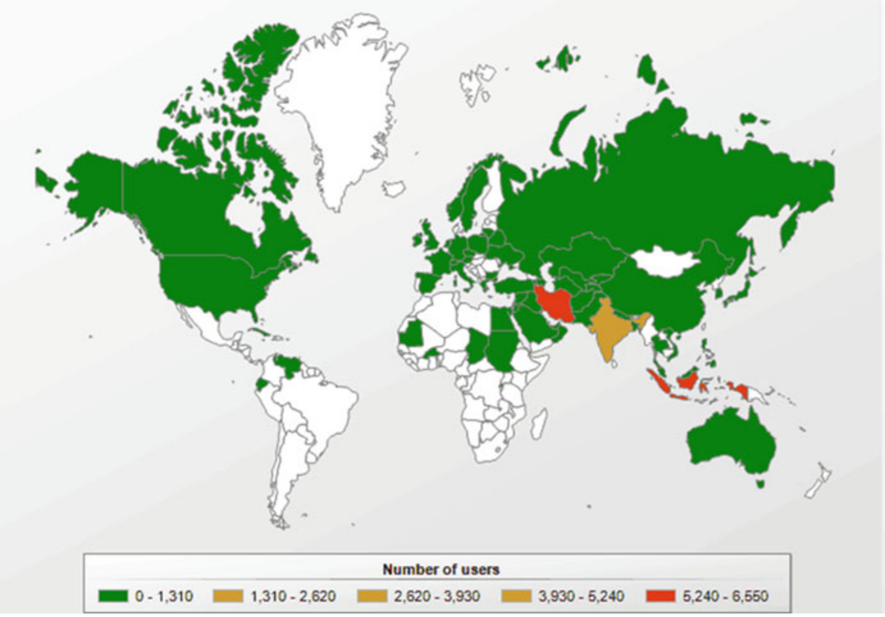


Fig. 1.1 Geography of users infected with the Rootkit.Win32.Stuxnet worm

MS10-061 Spool Server Exploit, MS10-073 Win32k.sys Exploit, MS10-092 Task Scheduler Exploit) [11]. Iranian security officials who dealt with *Stuxnet* indicate that the threat has not been entirely abolished since worms can have precise life cycles and remain active in their activities in other forms. They also highlighted that Iran should get ready itself to future worms' challenges, which may infect the country's infrastructure. *Stuxnet* has brought before the security community, a glaring possibility of a serious threat to any country's sovereignty. *Network forensics* is definitely one way to be prepared for such eventualities.

Infosecurity [12] reported that there is an increasing demand for network forensics as enterprises want to be sure about who, what, when, why, where, and how their services were being accessed and used. Network forensics cannot stop attacks like *Stuxnet*, but it can provide a way to decrease the impact by providing analysis that enables a more rapid response to the infection.

Solera Networks [13] explains that network forensics prepares organizations to respond swiftly to zero-day, negative day, and unknown threats. It boosts the value and effectiveness of other security investments. It reduces and simplifies the monitoring, reporting, analysis, and remediation time required to defend against attacks. It assists prosecution through evidence which is forensically complete and provides understanding of the root causes for the breach of security to enable rapid, intelligent, and effective response to prevent catastrophic events and ongoing risk. It allows for validation of fixes installed after a breach occurred through the ability to replay a network attack.

Network forensics appears to be similar to network security. However the objectives of the two are very much different. Network forensics is a nascent science that deals with capture, record, and analysis of network traffic. The network traffic data is captured using packet sniffers, and alerts and logs are collected from existing network security tools. This data is analyzed for attack characterization and investigated to trace back the perpetuators. This process can bring out deficiencies in security products which can be utilized to guide deployment and improvement of these tools.

The network security approach uses defensive mechanisms like firewalls and intrusion detection system (IDS). Firewalls are used for prevention and the IDS for detection. These approaches stereotypically find out network vulnerabilities and block all malicious communications from outside. Firewalls control traffic that enters a network and leaves a network, based on source and destination addresses and port numbers. It filters malicious network traffic according to the firewall rules. It is difficult to update the signatures of all vulnerabilities as new vulnerabilities will always keep occurring.

Intrusion detection system (IDS) [14] are primarily for learning, detecting, and reporting attacks as they happen in real time and have no evidence gathering feature. IDSs are of two types – signature-based (misuse) detection and statistical-based (anomaly) detection. Pattern matching is done in signature-based IDS to detect intrusion signatures. It cannot detect new attacks but has a low false positive rate. Anomaly-based IDS does activity monitoring and is able to detect new attacks but has higher false positive rate.

The network forensic approach collects the required evidence for incident response and investigation of the crime. Network security protects system against attack. Network security tools are generalized and continuously monitor the network for possible harmful behaviors. Network forensics encompasses postmortem investigation of the attack and is initiated *notitia criminis* (after crime notification). It is case specific as each crime scenario is different in many aspects, and the process is time bound. There may be certain crimes which do not breach network security policies but may be legally prosecutable. These crimes can be handled only by network forensics [15]. The major differences between network security and network forensics are given in Table 1.1.

Table 1.1 Comparison of network security and network forensics

Network security	Network forensics
System protection against attack	No system protection against attack
Usually in real time	Postmortem
Generalized – looking for any possible harmful behaviors	Case restricted – want to reconstruct the criminal scenario
Keep alert 24 h every day	After crime notification – notitia criminis
Continuous process	Time-bound process
Established field of computer science	Very immature and young science

Table 1.2 Comparison of computer forensics and network forensics

Computer forensics	Network forensics
Introduced by law enforcement to handle computer data	Evolved as a response to the hacker community
The investigator and attacker are on two different levels	The investigator and the attacker are at the same skill level
The investigator and attacker use different tools, investigator has upper hand	The investigator and attacker use same tools and practices
Computer forensics contains preservation, identification, extraction, documentation, and interpretation of data	Network forensics involves the capture, record, and analysis of network events
It is about acquiring, providing chain of custody, authenticating, and interpretation	It is about investigation of packet filters, firewalls logs, and IDS logs

Network forensics can be generally defined as a science of discovering and retrieving evidential information in a networked environment about a crime in such a way as to make it admissible in court [16]. The investigation of a cyber crime often involves cases related to homeland security, corporate espionage, child pornography, traditional crime assisted by computer and network technology, employee monitoring, or medical records, where privacy plays an important role.

Network forensics is a natural extension of computer forensics. Computer forensics [17] was introduced by law enforcement and has many guiding principles from the investigative methodology of judicial system. Computer forensics involves preservation, identification, extraction, documentation, and interpretation of computer data. Network forensics evolved as a response to the hacker community and involves capture, recording, and analysis of network events in order to discover the source of attacks.

In computer forensics, the investigator and the hacker being investigated are at two different levels with investigator at an advantage. In network forensics, the network investigator and the attacker are at the same skill level. The hacker uses a set of tools to launch the attack, and the network forensic specialist uses similar tools to investigate the attack. Network forensic investigator is more at a disadvantage, as investigation is one of the many jobs he is involved. The hacker has all the time at his disposal and will regularly enhance his skills, motivated by million dollars at stake. The seriousness of what is involved makes network forensics an important research field. The major differences between computer forensics and network forensics are given in Table 1.2.

1.2 Definition of Network Forensics

Network forensics deals with data which are found across a network connection mostly ingress and egress traffic. Network forensics attempts to analyze traffic data logged through firewalls or IDS or at network devices like routers and switches.

Network forensics is defined in [18] as “use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities.”

Ranum [19] defines network forensics as “capture, recording, and analysis of network events in order to discover the source of security attacks or other problem incidents.”

Network forensics comprises of network traffic monitoring and determining if there is an anomaly in the traffic and ascertaining whether it indicates an attack. If an attack is detected, next to determine the nature of the attack too. Techniques of network forensic empower investigators to trace back the attacker(s). The ultimate goal is to provide sufficient evidence to allow the perpetrator to be prosecuted [20].

1.3 Classification of Network Forensic Systems

Network forensic systems are classified into different types, based on various characteristics:

Purpose *General Network Forensics (GNF)* focuses on enhancing security. The network traffic data is analyzed, and attack patterns are discovered. *Strict Network Forensics (SNF)* involves rigid legal requirements as the results obtained will be used as evidence for prosecution of the network crimes [21].

Packet Capture *Catch-it-as-you-can* systems capture all packets passing through a particular traffic point and subsequently analyze them, requiring large amounts of storage. *Stop-look-and-listen* systems analyze each packet in memory, and only certain information is saved for future analysis, requiring a faster processor [22].

Platform Network forensic system can be a *hardware appliance with pre-installed software*. It can capture data, analyze it and present the results on a computer interface. It can also be standalone *software,* which can be installed on a host. It analyzes packet captures or NetFlow records, which are copied and stored in the host.

Time of Analysis Commercial network forensic analysis appliances involve *real-time* network surveillance, signature-based anomaly detection, data analysis, and forensic investigation. Many open-source software tools are designed for *post-mortem* investigation of packet captures. Full packet data is captured by sniffer tools, stored in a host and analyzed offline at a later time.

Data Source *Flow-based systems* collect statistical information based on some criteria within the network traffic as it passes through the network. The network equipment collects this data and sends it to a flow collector which stores and analyzes the data. *Packet-based systems* involve full packet captures at various points in the network. The packets are collected and stored for deep packet inspection.

1.4 Motivation

The real motivation for our present study comes directly from the limitations in the defensive approaches of network security like firewalls and intrusion detection systems. Firewalls and IDS can address attacks only from the perspectives of prevention, detection, and reaction. The alternative approach of network forensics becomes important as it involves the investigative component as well [23]. Network forensics ensures that an attacker spends more time and energy to cover his/her tracks, thus making the effort of an attack costly. Network criminals will be more cautious to avoid prosecution for their illegal actions. This acts as a deterrent and may reduce network crime rate, thereby improving security.

The large number of security incidents affecting many organizations and increasing sophistication of the cyber attacks is the main driving force behind network forensics. Successful attackers often ensure that they cover their trails. Unsuccessful attacks often go unnoticed, and little information is available to assist with diagnosis even when they are noticed [24]. Internet service providers (ISPs) are also being made responsible for what passes over their network [25]. Companies doing business on the Internet cannot hide a security breach and are now expected to prove the state of their security as a compliance measure for regulatory purposes.

The ISO 27001/27002 standard (information technology – security techniques – information security management) [26] specifies the requirements for establishing, implementing, operating, monitoring, reviewing, maintaining, and improving a documented information security management system (ISMS) within the context of the organization's overall business risks. Comprehensive audit data are to be maintained to meet the compliance requirements of many regulations.

Sarbanes–Oxley (SOX) Act controls over the release of information to individuals or organizations. Gramm–Leach–Bliley Act (GLBA) ensures the privacy and integrity of customer records. Health Insurance Portability and Accountability Act (HIPAA) was established to protect the health-related data. Federal Information Security Management Act (FISMA) monitors security programs for federal agencies. By adhering to the Payment Card Industry (PCI) Data Security Standard (DSS), retailers, service providers, and allied organizations can dramatically reduce the vulnerabilities that are easily exploited for the purpose of compromising corporate data. An integrated network forensic process will facilitate meeting compliance requirements [27] for organizations and ISPs by adhering to strict security measures and maintaining comprehensive audit data [28].

Network forensics also facilitates recording evidence for investigation and helps in understanding the attacker's methodology. It provides insight about the tools used by the attacker and new ways in which perimeter defenses were circumvented. This information can also bring to light the deficiencies in existing network security tools. These tools can be hardened to become robust enough to stand the onslaught of many zero-day and hybrid attacks.

1.5 Recent Trends in Network Forensics

Network forensics was traditionally applied to wired environments and was focused on the Version 4 of the Internet Protocol and related protocols at the network layer of the TCP/IP protocol suite. Following are some of the recent works in network forensics:

Steganography Many attackers use somewhat "light" forms of cryptography to render the recognition of rootkits or attack patterns to be more difficult, which otherwise would have been easily spotted by any IDS [29].

Honeypot Forensics Honeypots are placed to be compromised and provide information on the black hat's techniques and tools, before and after the intrusion on the honeypot. New forms of rootkits, trojans, and potential zero-day exploits can be discovered. A better understanding of the areas of interest and hidden links between black hat teams can be obtained [30, 31].

IP Version 6 Forensics IPv6 Internet provides malicious users a temporary safe haven, as events are poorly logged and monitored. Many free tunnel brokers provide simple and relatively anonymous connectivity [32]. The transition from IPv4 to IPv6 will take time, and both protocols may coexist for quite some time requiring interoperation mechanism. This dual-stack arrangement will bring new security vulnerabilities and exploits which will need forensic analysis [33].

Botnet Forensics Compromised machines can be linked up to form "botnets" under external control, which are used to send spam e-mails or disable Websites with a flood of bogus requests. It is very difficult to trace the identity of spammers by just analyzing the electronic trail [34, 35].

Wireless Network Forensics Companies are embracing wireless technology at a rapid pace, and the frequency of data leakage and theft is constantly increasing. There is a great need for profiling user activities emphasizing the need for 802.11 network monitoring and content inspection [36]. There is a clear lack of tools and procedures for forensic computing investigations to effectively handle wireless devices. Hence, there are many forms of misuse that escape detection [37].

VoIP over wireless (VoIPoW) networks are becoming the most popular system for mobile communication in the world. However, studies of attacks on wireless

VoIP networks are still in their infancy [38]. Challenges exist in Mobile Ad Hoc Networks (MANETs) where the number of the evidence packets is controlled by the level of reliability [39, 40].

Application Layer Forensics Attacks have moved from the network and transport layer to the application layer of the TCP/IP protocol suite. Attacks on Web security include cross site scripting (XSS), SQL injection, buffer overflows, etc. Reliable digital evidence can be provided form the payload of the network data traffic being transmitted to and from the Web service [41]. Domain name service forensics is also an important challenge [42].

SCADA Network Forensics Supervisory Control and Data Acquisition systems are widely used in industrial control and automation. Modern SCADA protocols often employ TCP/IP to transport sensor data and control signals. The use of TCP/IP as a carrier protocol and the interconnection of IT and SCADA networks raise serious security issues. Successful attacks on an IT network and its gateway devices could tunnel into a SCADA network, wreaking havoc on the industrial process [43, 44].

Grid Forensics Grid computing aggregates all kinds of heterogeneous resources that are geographically distributed and requires in-depth security services to protect its resources and data. It also entails suitable forensics techniques that can be employed to assess the responsibility of the wrongdoers. Security teams lack the experience of grid forensics as grid computing is itself, a growing technology [45].

Forensic Data Representation Garfinkel [46] predicts an impending crisis in digital forensics as many observers have identified the continuation of current trends. There is a serious need to make digital forensic research more efficient through the creation of new abstractions for data representation forensic processing.

Cloud Forensics Cloud computing will require a change in corporate and security policies concerning remote access, use of the data over a browser, privacy and audit mechanisms, reporting systems, and management systems that incorporate how data is secured on a rented computer system that can be anywhere in the world. The complex series of interlinkages between the cloud provider and the cloud consumer provides a fertile ground for hackers and criminals. Network forensics in cloud computing requires a new investigative mindset, where some data will not be available, some data will be suspect, and only some data will be court ready [47].

Intelligent Network Forensics An intelligent network forensic system reconstructs intrusion scenarios and makes attack attributions require knowledge about intrusions signatures, evidences, impacts, and objectives. Problem-solving knowledge that describes how the system can use domain knowledge to analyze malicious activities is essential. Saad and Traore [48] adapt recent researches in Semantic Web, information architecture, and ontology engineering to design method of ontology for network forensic analysis.

1.6 Challenges in Network Forensic Analysis

The challenges in network forensic analysis [49–51] are elaborated and classified based on the phases in the DFRWS model (2001):

Identification Attacks must be identified instantaneously and trigger the forensic process. The network events which are malicious must be identified. Future and zero-day attacks must be predicted based on the common attack features. The hacker groups have common types of attacking tools and the frequently utilized techniques.

Preservation Network traffic is very volatile (dynamic) and must be captured and preserved immediately, otherwise it is lost forever. Most network security tools do not produce hash values for captured data or utilize the same hash algorithms resulting in inconsistencies. Integrity of collected data has to be preserved so that the captured data will pass stringent legal procedures and qualify as evidence in a court of law.

Collection Capturing real-time network traffic transmitted throughout high-speed networks, without network traffic packets being dropped or lost, is an important challenge. Full packet captures will result in a very large amount of data. The process can be made efficient by collecting useful data only. Data collected may be reduced by filtering the data according to rules customized for a specific purpose. Network security devices must be able to handle unique input formats and produce different output formats. They must also facilitate universal time synchronization of display time in different formats and varying time zones between the devices.

Examination Packet captures are to be examined to identify protocol features which are manipulated. This information is correlated with attack events, and the compromise is validated. Validation of attack takes the process to the investigation phase. Packets are reorganized into individual transport-layer connections between machines, and the attack behavior is analyzed by replaying the attack.

Analysis End-to-end and link encryption technology prevents captured network traffic from being analyzed. Logging data from different locations can give reconnaissance of the attacking behavior. The analysis of the aggregation of the data sets, which are from multiple sources, such as firewalls, IDSs, and sniffers, can build the chain of the clues and display the full scene of the crime.

Presentation Many network security devices do not have the ability for visually analyzing the network traffic and log data. Documentation needs to be done for every step in order to ensure that all precautions have been taken and that no privacy violations have taken place.

Decision (Investigation) IP trace back methods can trace a steady stream of anonymous Internet packets back toward their source to the attack origin. These methods do not rely on knowledge or cooperation from intervening ISPs along the path. The attacker may launch the attack in a very short time and use only a few packets making the trace back process difficult.

1.7 Conclusion

Network forensics was introduced and compared with network security and computer forensics. Network forensics was also formally defined, and various types of network forensic systems were classified. The motivation for research in network forensics is elucidated. The recent trends in this nascent field are explored, and the research challenges in network forensics are highlighted.

1.8 Questions

Objective Questions

1. The term network forensics was coined by ________________________
2. ________________________ is an open-source network forensic tool.
3. ________________________ systems capture all packets passing through a particular traffic point and subsequently analyze them.
4. FISMA deals with ________________________
5. Latest domain to which forensics is applied is ________________________

Short-Answer Questions

1. Compare computer forensics and network forensics.
2. Find the similarities in network security and network forensics.
3. What are the challenges in wireless network forensics?
4. What are the various classifications of network forensics?
5. What is compliance? How does network forensics help in meeting the requirements?

Long-Answer Questions

1. Discuss new areas of networking, in which forensics has made in roads. Identify research challenges while bringing out the differences in traditional approach.
2. Examine the challenges in network forensics and propose possible solutions to overcome them.

References

1. LA Times. Hack at Sony Pictures shuts computer system. Available: http://www.latimes.com/entertainment/envelope/cotown/la-fi-sony-hack-20141125-story.html. [31 Mar 2016]
2. Sony breach linked to Romania. Available: https://sony.attributed.to. [31 Mar 2016]
3. Schneier B. Attributing the Sony Attack. Available: https://www.schneier.com/blog/archives/2015/01/attributing_the.html. [31 Mar 2016]
4. NY Times. Chinese army unit is seen as tied to hacking against U.S. Available: http://www.nytimes.com/2013/05/20/world/asia/chinese-hackers-resume-attacks-on-us-targets.html?_r=0. [31 Mar 2016]
5. NY Times. Hackers from China resume attacks on U.S. targets. Available: http://www.nytimes.com/2013/02/19/technology/chinas-army-is-seen-as-tied-to-hacking-against-us.html. [31 Mar 2016]
6. Desk TTP (2011va) New cyber attack targets Iran. Available: http://www.tehrantimes.com/PDF/11135/11135-1.pdf. [31 Mar 2016]
7. Falliere N, Murchu LO, Chien E (2011) W32. Stuxnet Dossier. Available: http://www.symantec.com/en/ca/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf. [31 Mar 2016]
8. deSouza F (2011) Safeguarding critical infrastructure from the next Stuxnet. Available: http://www.networkworld.com/news/tech/2011/042711-infrastructure-stuxnet-safeguard.html. [31 Mar 2016]
9. Mills E, Langner R (2011) Ralph Langner on Stuxnet, copycat threats (Q&A). Available: http://news.cnet.com/8301-27080_3-20061256-245.html. [31 Mar 2016]
10. Kessler G (2010) Stuxnet worm possibly made to cripple Iran centrifuges. Available: http://www.washingtonpost.com/wp-dyn/content/article/2010/11/15/AR201011.1506.768.html. [31 Mar 2016]
11. Matrosov A, Rodionov E, Harley D, Malcho J (2011) Stuxnet under the microscope. Available: http://www.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf. [31 Mar 2016]
12. Infosecurity (2010) Network forensics helps bolsters confidence in cloud computing security. Available: http://www.infosecurity-us.com/view/13252/network-forensics-helps-bolsters-confidence-in-cloud-computing-security/. [31 Mar 2016]
13. Networks S (2010) Unveiling the security illusion. Available: http://www.soleranetworks.com/resources/wp_need_for_net_forens_web.pdf. [31 Mar 2016]
14. Axelsson S (2000) Intrusion detection systems: a survey and taxonomy. Department of Computer Engineering, Chalmers University, Gothenburg, Technical Report 99–15, March 14, 2000
15. Broucek V, Turner P (2001) Forensic computing: developing a conceptual approach for an emerging academic discipline, In: 5th Australian Security Research symposium, Perth, Australia, pp 55–68
16. Guan Y (2009) Network forensics. In: John RV (ed) Computer and information security handbook. Morgan Kaufmann, Boston, pp 339–347
17. Berghel H (2003) The discipline of internet forensics. Commun ACM 46(8):15–20
18. Palmer G (2001) Digital Forensic Science in Networked Environments (Network Forensics). In: 1st Digital Forensic Research Workshop (DFRWS' 01), Utica, New York, USA, pp 27–30
19. Ranum MJ (1999) Intrusion detection and network forensics. In: 2nd USENIX symposium on Internet Technologies and Systems, Colorado, USA
20. Yasinsac A, Manzano Y (2001) Policies to Enhance Computer and Network Forensics. In: IEEE workshop on Information Assurance and Security, New York, USA, pp 289–295
21. Ren W, Jin H (2005) Modeling the network forensics behaviors. In: Workshop of the 1st international conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm' 05), Athens, Greece, pp 1–8

22. Garfinkel S (2002) Network forensics: tapping the internet, Available: http://www.oreillynet.com/pub/a/network/2002/04/26/nettap.html. [31 Mar 2016]
23. 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
24. Laurie B (2004) Network forensics. ACM Queue 2(4):50–56
25. Perry S (2006) Network forensics and the inside job. Netw Secur 2006(12):11–13
26. ISO (2005) ISO/IEC 27001:2005 information technology – security techniques – information security management systems – requirements. Available: http://www.iso.org/iso/catalogue_detail.htm?csnumber=42103. [31 Mar 2016]
27. Haugdahl JS (2007) Network forensics: methods, requirements, and tools. Available: http://www.bitcricket.com/downloads/Network%20Forensics.pdf. [31 Mar 2016]
28. netForensics (2010) Security compliance management. Available: http://www.netforensics.com/compliance/. [31 Mar 2016]
29. Forte D (2002) The future of computer and network forensics. Netw Secur 2002(10):13–15
30. Raynal F, Berthier Y, Biondi P, Kaminsky D (2004) Honeypot forensics, Part 1: analyzing the network. IEEE Secur Priv 2(4):72–78
31. Raynal F, Berthier Y, Biondi P, Kaminsky D (2004) Honeypot forensics, Part II: analyzing the compromised host. IEEE Secur Priv 2(5):77–80
32. Nikkel BJ (2007) An introduction to investigating IPv6 networks. Digit Investig 4(2):59–67
33. Govil J, Govil J, Kaur N, Kaur H (2008) An examination of IPv4 and IPv6 networks: constraints and various transition mechanisms, In: IEEE Southeastcon 08, Huntsville, Alabama, USA, pp 178–185
34. Vural I, Venter H (2010) Mobile botnet detection using network forensics. In: Berre A, Gómez-Pérez A, Tutschku K, Fensel D (eds) Future internet – FIS 2010, vol 6369. Springer, Berlin/Heidelberg, pp 57–67
35. Vural I, Venter HS (2010) Using network forensics and artificial intelligence techniques to detect bot-nets on an organizational network. In: Seventh international conference on Information Technology: New Generations (ITNG' 10), Las Vegas, Nevada, USA, pp 725–731
36. Qureshi A (2009) 802.11 Network forensic analysis. Available: http://www.sans.org/reading_room/whitepapers/wireless/80211-network-forensic-analysis_33023. [31 Mar 2016]
37. Turnbull B, Slay J (2008) Wi-Fi network signals as a source of digital evidence: wireless network forensics. In: Third international conference on Availability, Reliability and Security (ARES 08), Barcelona, Spain, pp 1355–1360
38. Pelaez JC, Fernandez EB (2006) Wireless VoIP network forensics. In: Fourth LACCEI international Latin American and Caribbean conference for Engineering and Technology (LACCET' 06), Mayagüez, Puerto Rico, pp 1–12
39. Otaka A, Takagi T, Takahashi O (2008) Network forensics on mobile Ad-Hoc networks. In: Lovrek I, Howlett R, Jain L (eds) Knowledge-based intelligent information and engineering systems, vol 5179. Springer, Berlin/Heidelberg, pp 175–182
40. Yinghua G, Simon M (2010) Network forensics in MANET: traffic analysis of source spoofed DoS attacks. In: 4th international conference on Network and System Security (NSS' 10), Melbourne, Australia, pp 128–135
41. Guo R, Cao T, Luo X (2010) Application layer information forensics based on packet analysis. In: International conference of Information Science and Management Engineering (ISME' 10), Xian, China, pp 206–209
42. Nikkel BJ (2004) Domain name forensics: a systematic approach to investigating an internet presence. Digit Investig 1(4):247–255
43. Kilpatrick T, Gonzalez J, Chandia R, Papa M, Shenoi S (2008) Forensic analysis of SCADA systems and networks. Int J Secur Netw 3(2):95–102
44. Kilpatrick T, Gonzalez J, Chandia R, Papa M, Shenoi S (2006) An architecture for SCADA network forensics. In: Olivier M, Shenoi S (eds) Advances in digital forensics II, vol 222. Springer, Boston, pp 273–285

45. Naqvi S, Massonet P, Arenas A (2006) Scope of forensics in grid computing – vision and perspectives. In: Min G, Di Martino B, Yang L, Guo M, Rünger G (eds) ISPA’ 06 workshop on frontiers of high performance computing and networking, vol 4331. Springer, Berlin/Heidelberg, pp 964–970
46. Garfinkel SL (2010) Digital forensics research: the next 10 years. Digit Investig 7(Supplement 1):S64–S73
47. Lillard TV, Garrison CP, Schiller CA, Steele J (2010) What is network forensics?. In: Digital forensics for network, internet, and cloud computing. Syngress, Boston, pp 3–20
48. Saad S, Traore I (2010) Method ontology for intelligent network forensics analysis. In: Eighth annual international conference on Privacy Security and Trust (PST’ 10), Ottawa, Ontario, Canada, pp 7–14
49. Almulhem A (2009) Network forensics: notions and challenges. In: IEEE International Symposium on Signal Processing and Information Technology (ISSPIT’ 09), Ajman, UAE, pp 463–466
50. 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
51. Lillard TV, Garrison CP, Schiller CA, Steele J (2010) The future of network forensics. In: Digital forensics for network, internet, and cloud computing. Syngress, Boston, pp 341–347

Chapter 2
Network Forensic Process Models

Learning Objectives

- Background of various process models in digital forensics and network 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

There exist some investigative techniques and methods for the traditional computer forensic discipline which have been validated and justified. However as we have become more and more networked and use mobile at home and business, there is a need to expand our forensic view from disk level to the network level. There is a need to factor this transition into concepts, designs, and prototypes. Various digital forensic models were proposed to handle the networked environments since 2001. The term "model" is used to imply a theoretical representation of phases involved in network forensics. The model may or may not have been implemented. Main focus of earlier digital forensic process model is to focus on investigation of a stand-alone computer and interpret the stored data. The forensic experts have the knowledge of specialized tools, which the attackers lack. As a response to hackers' community, network forensics is evolved. All the process models are described in various phases which show various kinds of activities carried out during investigation. The phases have been evolved and improved in a last few years of research to increase the accuracy and efficiency of investigation. With the advancement in technology and tools, investigation output has also been improved. Mainly there are two categories of process models (a) digital forensic process models (b) network forensic process models as explained below:

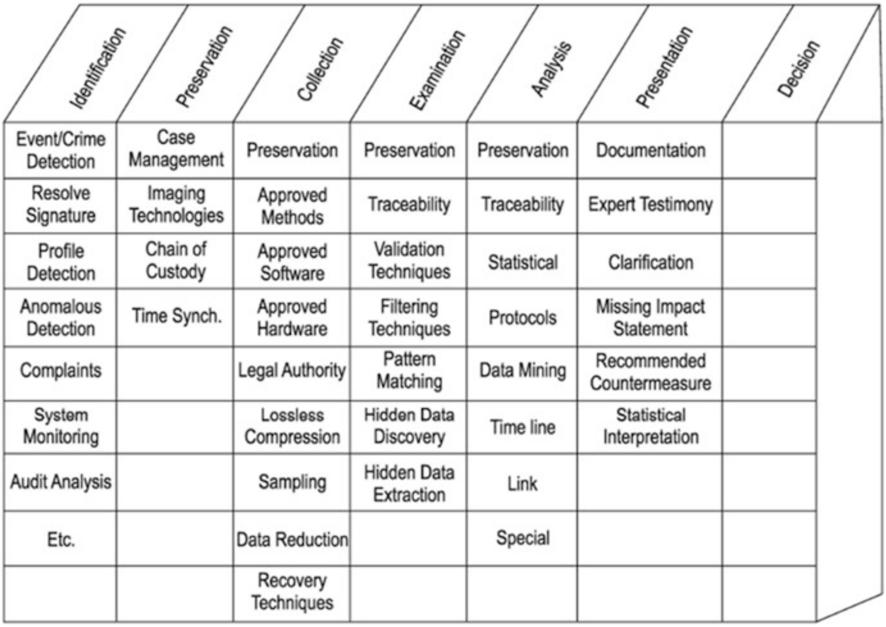
R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer
Communications and Networks, DOI 10.1007/978-1-4471-7299-4_2

2.1 Digital Forensic Process Models

Digital forensics [1] has been defined as "The use of scientifically derived and proven methods towards the preservation, collection, validation, identification, analysis, interpretation and presentation of digital evidence derived from digital sources. Digital evidence includes computer evidence, digital audio, digital video, cell phones, digital fax machines etc." Various digital forensic process models are explained below:

The first Digital Forensic Research Workshop (DFRWS) took the first attempt to apply digital forensic science to networked environments as one of the objectives in 2001, and the framework is shown in Fig. 2.1.

There are following steps in the framework: identification, preservation, collection, examination, analysis, presentation, and decision. Top row of the table shows the major classes. Contents below the column are the methods or techniques used for that class. Here real-time analysis must be considered as an essential research objective. For carrying out analysis, repository of digital forensic knowledge must be constructed. Collaborative technologies are helpful to perform the forensic investigation. The process is presented as linear, though the feedback must be incorporated in order to make it effective. Real-time analysis is done to make the detection most effective.



Identification	Preservation	Collection	Examination	Analysis	Presentation	Decision
Event/Crime Detection	Case Management	Preservation	Preservation	Preservation	Documentation	
Resolve Signature	Imaging Technologies	Approved Methods	Traceability	Traceability	Expert Testimony	
Profile Detection	Chain of Custody	Approved Software	Validation Techniques	Statistical	Clarification	
Anomalous Detection	Time Synch.	Approved Hardware	Filtering Techniques	Protocols	Missing Impact Statement	
Complaints		Legal Authority	Pattern Matching	Data Mining	Recommended Countermeasure	
System Monitoring		Lossless Compression	Hidden Data Discovery	Time line	Statistical Interpretation	
Audit Analysis		Sampling	Hidden Data Extraction	Link		
Etc.		Data Reduction		Special		
		Recovery Techniques				

Fig. 2.1 Investigative process of digital forensic science

Carrier and Spafford [2] proposed an integrated digital investigation process based on the approaches of physical investigations as shown in Fig. 2.2. Readiness phase ensured operations infrastructure is geared up. Readiness phase ensures that operations and the infrastructure fully support the investigation. Operational readiness provides training and equipment for the personnel which will be used during investigation. Infrastructure readiness ensures whether needed data exists or not. The search and collection phases gather and process the data. Reconstruction and analysis phase are quite similar to each other. The documentation phase records all the evidence. Deployment phase provides the mechanism for detection of incident. Various activities of physical crime scene investigation are shown in Fig. 2.3. The major goal of this phase is collecting and analyzing the evidences. The preservation phase includes activities such as detaining suspects, identifying witnesses, helping the wounded, and securing the evidence. The survey phase involves a walk-through of the scene by investigator. This is followed by documentation which involves taking photographs, sketches, video, etc. The next step is search and collection which involve representing the deep search in form of physical evidences. Survey results are used to focus on analysis types. The user activity can be traced by making timeline of file activity. The results are organized forming theory of

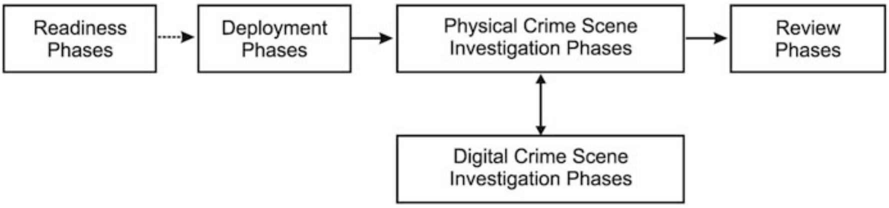

Readiness Phases
Deployment Phases
Physical Crime Scene Investigation Phases
Review Phases
Digital Crime Scene Investigation Phases

Fig. 2.2 Phases in investigation process

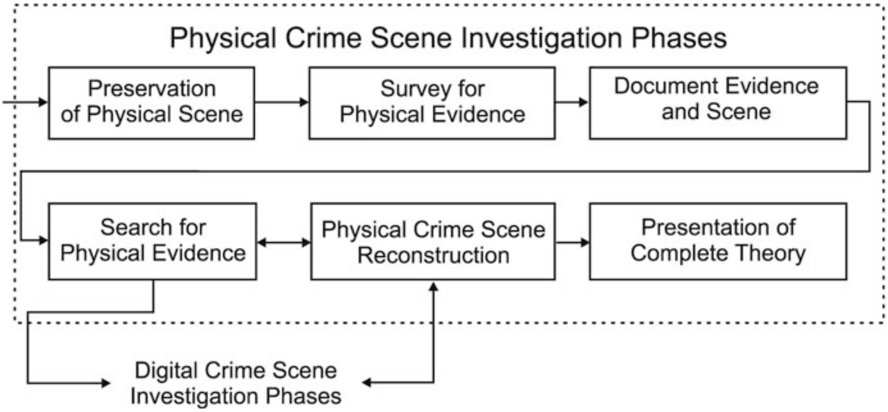

Physical Crime Scene Investigation Phases
Preservation of Physical Scene
Survey for Physical Evidence
Document Evidence and Scene
Search for Physical Evidence
Physical Crime Scene Reconstruction
Presentation of Complete Theory
Digital Crime Scene Investigation Phases

Fig. 2.3 Phases in the physical crime scene investigation and the interaction with the digital crime scene investigation

incidents. Representation of digital crime involves representing digital evidences that were found by investigating team. Digital crime investigation uses computer based approach and searches for evidence. Knowledge of techniques and tools is required to carry out digital investigation. Each digital device is considered as a separate crime scene. Finally investigation is reviewed for further improvement. The outcome of this phase could be new processes, training, etc.

Baryamureeba and Tushabe [3] proposed an enhanced version of integrated digital investigation process model which refines the phases and reorganizes the phases presented in [2]. The model is based on the physical crime investigation process. Two new phases, namely, traceback and dynamite, are included in the process. They have added new phases such as investigation, authorization, reconstruction, and communication giving clarity and granularity to the major phases. Development starts from the very beginning when readiness is confirmed. The appropriate people are notified on detection of an incident. Confirmation and authorization are other tasks of deployment phase. In traceback phase, physical crime scenes are traced down to identification of devices that are used in investigation. Dynamite phase investigates primary crime scene. The subphases are physical crime investigation phase, digital crime scene investigation phase, reconstruction phase, and communication phase. This model is suitable for crime investigation as shown in Fig. 2.4.

Casey and Palmer [4] proposed an investigative process model to ensure proper evidence handling and insist a complete rigorous investigation which reduces the chances of mistakes. Other than the common phase, assessment phase validates the incident, and a decision is provided whether to continue with the investigation.

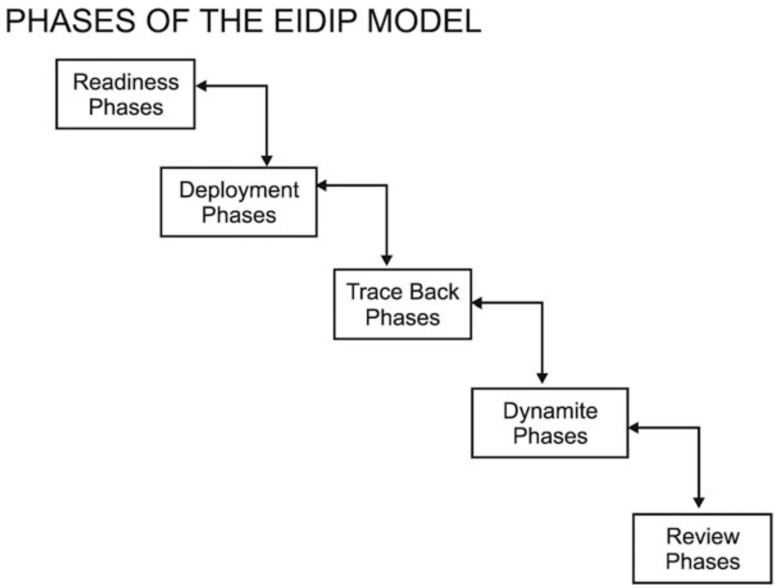


Fig. 2.4 Phases of proposed model

Harvesting, reduction, organization, and search phases rearrange the data so that a smallest set data with high potential evidence is generated. Persuasion and testimony phases present the case in common man terminology. Investigators work from bottom to top in a systematic way and at the end present the compiled story to the concerned authority. Output of one phase is transformed to another phase to make the process more focused and accurate. Case management plays an important role in entire phases which ensures stability and makes the investigators eligible to tie all information of all phases.

Analysis is carried out in each phase using scientific methods to validate the information acquired. The final results of analysis are presented in terms of report. The report is then presented to the authorized people. The report may be further translated and explained to the concerned authorities. Various phases of investigation are shown in Fig. 2.5.

Ieong [5] proposed a digital forensic investigation framework, FORZA, which incorporates legal issues. The author identifies eight roles to fulfill the fundamental principles for digital forensic investigation, namely, reconnaissance, reliability, and relevancy. He lists six questions for each role – what, why, how, who, where, and when. These roles and questions are incorporated into the Zachman's framework for enterprise architecture, and FORZA is composed. This model is being automated by developing a data acquisition script generator which will collect relevant information from the network logs. FORZA framework incorporates six sets of questions what, why, how, who, where, when, etc., to form investigation process. It

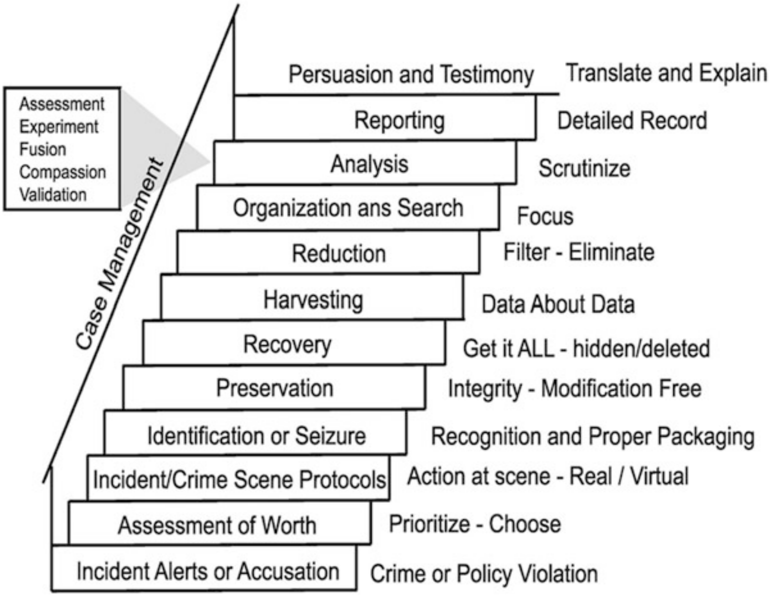


Fig. 2.5 Investigation process model

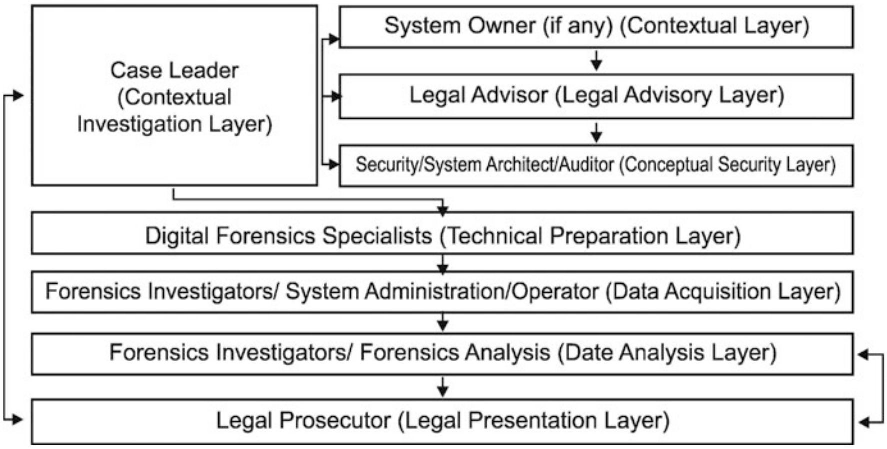

Case Leader (Contextual Investigation Layer)
System Owner (if any) (Contextual Layer)
Legal Advisor (Legal Advisory Layer)
Security/System Architect/Auditor (Conceptual Security Layer)
Digital Forensics Specialists (Technical Preparation Layer)
Forensics Investigators/ System Administration/Operator (Data Acquisition Layer)
Forensics Investigators/ Forensics Analysis (Date Analysis Layer)
Legal Prosecutor (Legal Presentation Layer)

Fig. 2.6 Digital forensic investigation framework

explains how legal advisors and prosecutors can incorporate into a digital forensic investigation. Various layers of the framework are depicted in the Fig. 2.6.

Selamat et al. [6] provided a mapping process in digital forensic investigation framework (DFIF) which is between the activities and output for each phase. A study of the existing digital forensic frameworks is done. A mapping is thereafter constructed. The same activities or processes are grouped and merged together to provide the same output into an appropriate phase. A mapping process is designed, that can produce concrete evidence to be presented in a court of law, to balance the process on achieving the overriding goal. The phases are preparation, collection, preservation, examination, analysis, presentation, reporting, and dissemination of the case. A simplified DFIF is developed by the use of mapping process which establishes the clear idea or approach to be followed while investigating. The proposed map can be further mapped to a number of incident cases to optimize investigation. Various phases such as acquisition, identification, evaluation, admission as evidence, analysis, dissemination, and presentation are mapped. The proposed map can be further mapped to a number of incident cases.

Grober [7] proposed a Digital Forensic Management Framework (DFMF) divided into three components: ProDF (proactive), ReDF (reactive), and ActDF (active). Proactive means creating or controlling a situation instead of just responding it. It reconstructs the technologies, processes, and procedures to create, collect, preserve, and manage Comprehensive Digital Evidence (CDE) to provide the cost-effective and successful investigation. ReDF involves the investigation that is conducted after the incident was detected and successful investigation. ReDF involves the investigation that is conducted after an incident was detected and confirmed. Various phases of ReDF are shown in Fig. 2.7. The first phase includes various steps: activity detection, reporting of an incident, assessment of the results, confirmation of the incident, formulation of hypothesis, obtaining authorization,

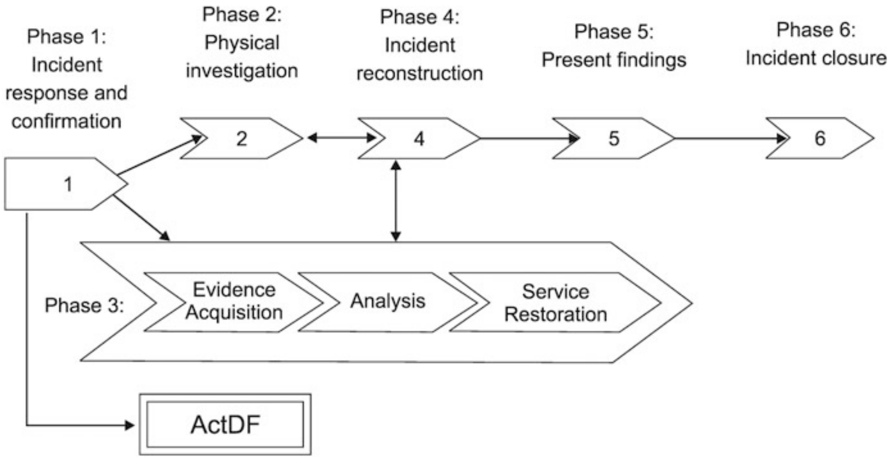


Fig. 2.7 ReDF phases

determination of containment strategy, making plan of investigation, coordinating the resources, and notification of investment. Next phase is physical investigation and then digital investigation which includes evidence acquisition, analysis, and service restoration. Phase four is incident reconstruction. Here the digital investigation findings and physical investigations are consolidated. Phase two and three can be repeated for improvement. Results are well documented. Next phase is to present the findings to management or authorities and other audiences considering legal jurisdiction location requirement. The third component, ActDF has the following phases. The first phase is incident response and confirmation to investigate the incident. The second phase is ActDF investigation followed by Event reconstruction and ActDF termination (Incident Closure).

The proposed model of Ademu [8] provides assistance to the forensic investigator to provide precise, authenticated, and accurate evidence to present to the court. The investigation process model consists of a four-tier iterative approach. The preparation and inception phase is the first-tier phase which iterates from the course of investigation to the final presentation. Rules for the first tier are preparation, identification, authorization, and communication. It consists of preparation of tools, techniques, search warrants, etc. Investigators need to be prepared with all these things before carrying out the investigation. Approaches and procedures are formulated in order to maximize the collection of evidence and to minimize the impact to victim. Rules for the second tier are collection, preservation, and documentation. Physical scenes and duplicate digital evidences and all other possible evidences are collected using standard procedures. All supporting evidences need to be preserved. Devices are put in envelopes and sealed before packaging. The preserved evidences are kept in a secure place and restricted from any unwanted access. All evidences help in the future investigations. The third tier consists of rules for examination, exploratory testing, and analysis. The information collected is examined and filtered

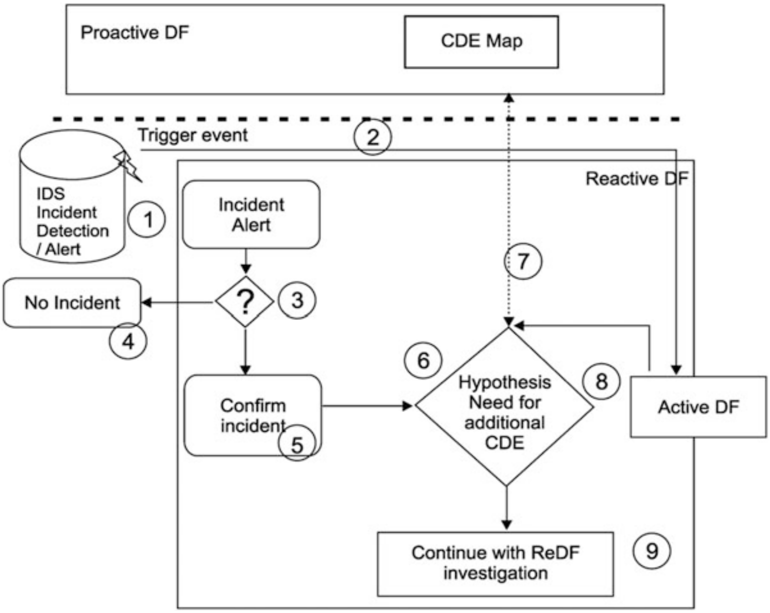


Fig. 2.8 Relationship between DF components

to form the required output. It is tested against the supporting evidences for the accuracy and validity, and finally the filtered information is analyzed, and outcomes are documented. The presentation in the fourth tier incorporates rules for result, review, and report. The report is presented in front of the court which includes summary and explanations (Fig. 2.8).

Agarwal [9] provides systematic model where the first phase is the preparation phase which involves collecting various materials for packing evidence sources as shown in Fig. 2.9. The investigation should be done under various legal constraints and jurisdiction and organization constraint as well. A strategy is developed for doing the investigation keeping in mind various legal, technical, and business factors. The second stage is securing the crime scene from unauthorized access. The third stage involves survey and recognition. Interviews are conducted for accumulating the evidences without violating the legal laws. Searching is done according to warrant only. An initial plan must be developed for collecting and analyzing evidences at the end of this phase. Phase four is documenting the scene with all supporting files such as sketching, photographing, and mapping of crime scenes. A documentation is maintained for all electronic devices at the scene. The next step is communication shielding. The devices are blocked which have been used for investigation so that existing information cannot be overwritten.

The next phase is evidence collection. The seventh phase talks about preservation which includes packaging, transportation, and storage. All the sources are identified and label them before packing. The devices are put in envelopes and sealed before

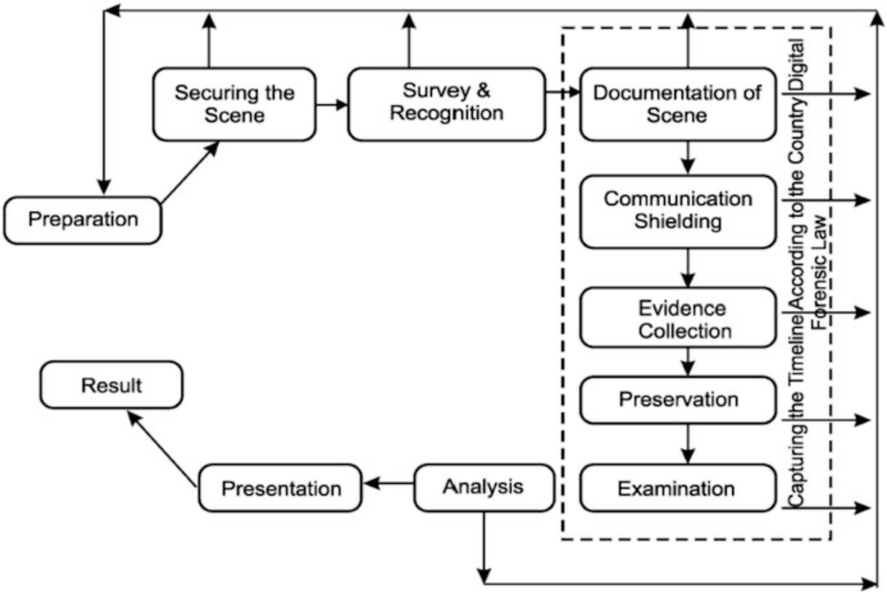


Fig. 2.9 Systematic digital forensic investigation model

placing in bag. The eighth phase is examination. Forensic analysis is done on the collected evidences. Data backup, data filtering, validation, pattern matching, and keyword searching are done during this phase. The next phase is about analysis. The investigation team conducts a technical review on the basis of the output of the examination phase. The information obtained from examination is analyzed to recognize the hidden pattern in data, and the relationship among the fragments of data is established. The tenth phase is presentation phase where results are presented before the audience including technical experts, legal experts, the corporate management, law enforcement officials, etc.

James and Gladyshev [10] surveyed fourteen digital forensic investigators to know about the investigation process used when child exploitation material (CEM) requests are investigated. Child exploitation material (CEM) was found to be the most critical crime to the risk to the victim. The observed investigation process for CEM consists of five main phases, namely, physical trail of material under suspense, preliminary analysis, social analysis, situational analysis, and full analysis, as shown in Fig. 2.10. The first three phases are always carried out. On the basis of the output, a decision is made to proceed for further investigation. The first phase is physical trial of suspect exhibits. The required evidences are collected from local members and any other documents submitted are sorted based on the attributes like owner, location, etc. Then the suspect device is acquired and a full examination is done. The computer forensic tool, EnCase [http://www.guidancesoftware.com/encase-forensic] is used for manual images and videos followed by file signature analysis, deleted file recovery, folder recovery, and keyword searching. Investigators look for

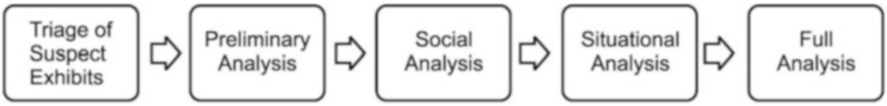


Fig. 2.10 High-level investigation process of child exploitation material

suspicious softwares and Internet history. Investigators can run the CEM detection software to hash and analyze the content of allocated and unallocated files. Further social analysis is carried out. The background information of suspect is collected to build a suspect profile like age, living situation, criminal record, etc. The digital user profile can also be considered. The next phase is situational analysis, where Chart logs are parsed and e-mails in CEM investigation are manually examined. The Windows Registry is also examined to determine the connected USB devices. Full analysis is carried out at the end which consists of all previously mentioned steps. Full analysis acquisition of the device is done during full analysis. In-depth analysis is carried out on the suspicious items found in the previous steps. Any additional step could be carried out to refine the investigation.

Shrivastava [11] proposes a digital forensic investigation model (DFIM) as a technique for computer examiners or investigators for carrying out the investigation. Initially we have to carry risk assessment followed by acquisition. The original documents are preserved. Various acquisition subfunctions include making a physical and logical copy of data, formatting data in a common format, and acquiring data through command line or GUI. The acquired data is validated and checked for the integrity. There are some subfunctions such as hashing, filtering, and analyzing file headers. Extraction is the next phase which is the recovery task in digital investigation used to analyze the investigated data. There are some extraction techniques such as keyword searching, decryption, decompression, bookmarking, etc. Next phase is reconstruction to recreate a suspect's storage drive image. The subfunctions for reconstruction phase are image to image copy, disk to disk copy, partition to partition copy, and image to partition copy. The log files will report the entire findings. The report is presented in the court in front of concerned authority as evidence. At the end, if there is no useful information in the retired system, disposition of the retired system is done so that it cannot be used for any malicious activity.

Kohn proposed [12] integrated digital forensic process model or IDFPM which is based on the following phases: preparation, incident response, physical investigation, digital forensic investigation, and presentation. The investigation starts with preparation in which policy and procedures are documented which ensures that the chain of evidence is constructed and maintained. Documentation is the continuous process of investigation. There should be accurate recording of physical scenes to help the digital and physical forensic investigators. Documentation includes techniques, procedures, and devices used in the investigation. The preparation process includes the operational and infrastructural readiness. To start an incident response, a well-defined policy should be priorly notified. The concerned authorities should grant the permissions to continue with the investigation like a police warrant or other authorizations, such as an attorney's. Further, we start with the incident response. The first responder should be able to elucidate various scenes in drafting of documentation using videos, sketches, and photographs. Interviews are conducted with witnesses and suspects, and a brief strategy is formulated to initialize the chain of custody and a robust chain of evidence. Finally digital forensic investigation is performed to find out the source of crime and type of crime. The process carried out in all phases is shown in Fig. 2.11. At the end a report is presented in the court as shown in Fig. 2.12.

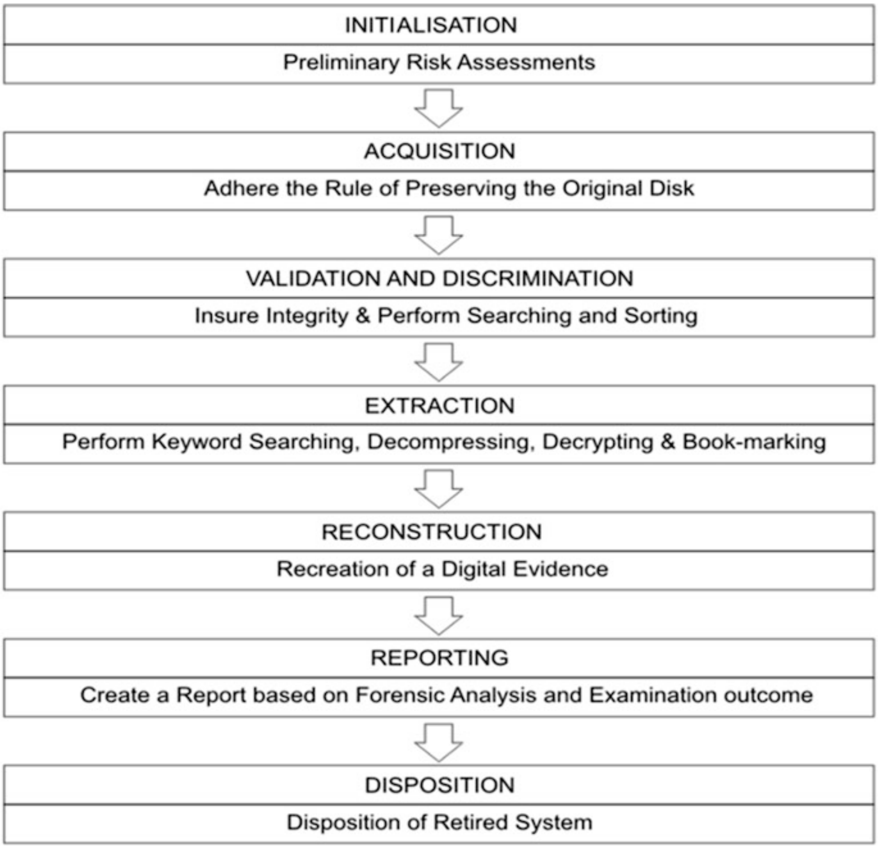

INITIALISATION
Preliminary Risk Assessments
ACQUISITION
Adhere the Rule of Preserving the Original Disk
VALIDATION AND DISCRIMINATION
Insure Integrity & Perform Searching and Sorting
EXTRACTION
Perform Keyword Searching, Decompressing, Decrypting & Book-marking
RECONSTRUCTION
Recreation of a Digital Evidence
REPORTING
Create a Report based on Forensic Analysis and Examination outcome
DISPOSITION
Disposition of Retired System

Fig. 2.11 Investigation process of DFIDM

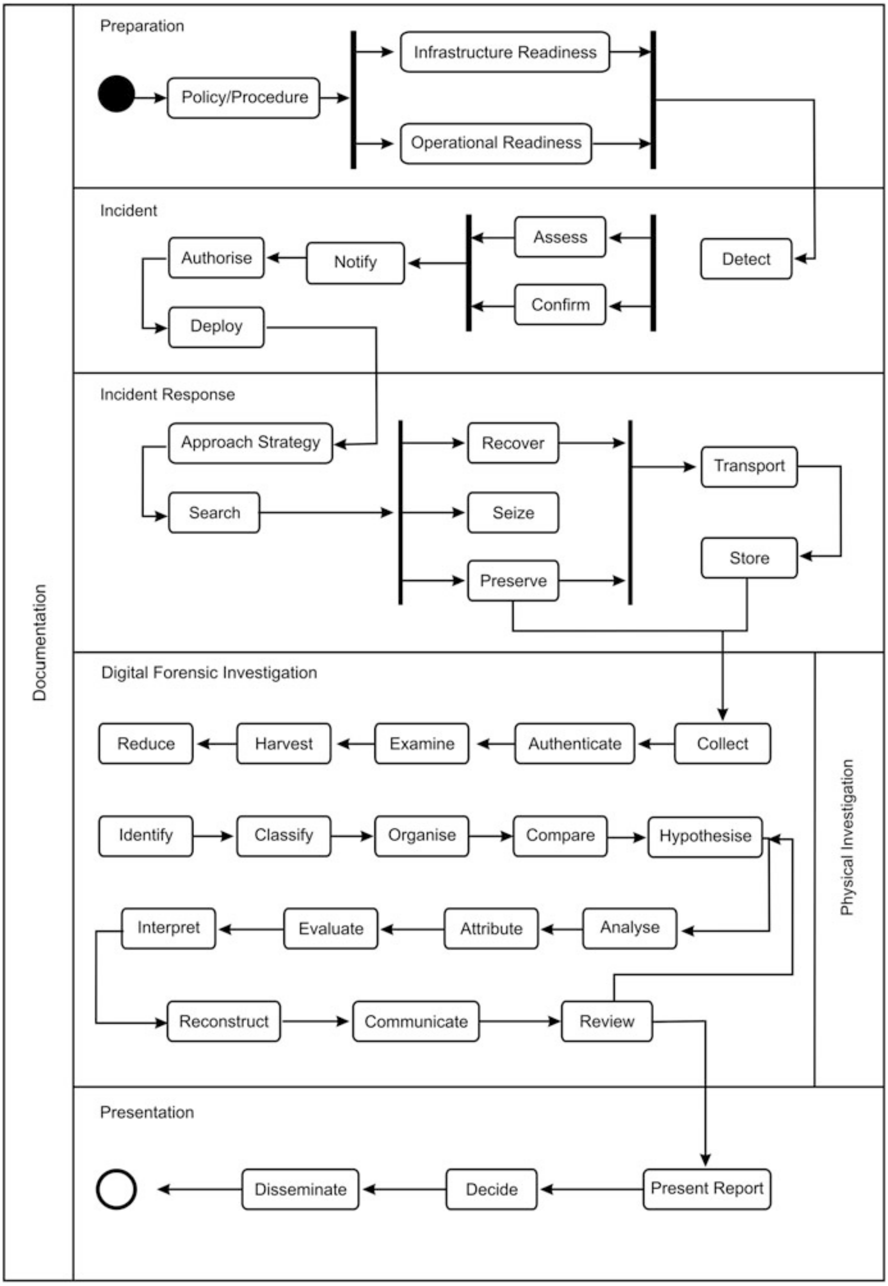


Fig. 2.12 Process flow of IDFMO model

2.2 Hierarchical Process Model

Beebe and Clarke [13] proposed a multi-tier, objectives-based hierarchical framework for a digital investigative process. There are common phases in the first tier, providing simplified view and a conceptual understanding. These common phases consist of subphases to provide specificity and granularity, guided by principles and objectives. The subphase structure for the data analysis phase was presented and analyzed.

First-Tier Phases In the first tier, preparation involves risk assessment, developing strategy and policy, preparing host and network devices and tools, developing legal activities, etc. The incident response phase is accountable for detecting, validating, assessing, and determining a response strategy for the security incident. Data is collected to validate the incident and determine its impact. Data analysis is carried out to reconstruct the events. It employs extraction techniques. Incident closure attempts to preserve knowledge gained to enhance subsequent investigation. At the end findings are presented. The first-tier framework is shown in Fig. 2.13.

The explanation of the first-tier phase is explained below.

Preparation

The preparation involves various activities such as risk assessment; developing an information retention plan; developing an incident response plan including policies, procedures, and personnel assignments; training of personnel; and training host and network devices. Preparation improves the quality and availability of digital evidences. It also takes care of evidence-handling procedures and developing legal activity coordination plan for both post- and pre-incidents.

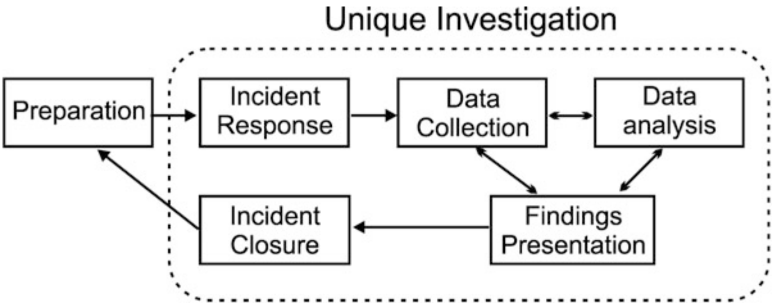


Fig. 2.13 First-tier phases of framework

Incident Response Phase

This phase is responsible for detecting a suspected computer crime-related incidents and initializing pre-investigation response. The prime aim is to identify, validate, assess, and determine a response plan for the suspected security incidents. Other various activities carried out in this phase are developing a plan regarding containment, elimination, recovery, and investigation; coordinating human, legal, and law enforcement resources; and formulating the appropriate investigation.

Data Collection Phase

The primary motive of this phase is to collect the digital evidence to support the response strategy and investigative plan. The data collection activities obtain network-based evidence from intrusion detection systems, firewall, log servers, etc. They acquire host-based evidence from relevant sources such as hard drive, system date/time, volatile data, etc.; install activity-monitoring capabilities such as network monitors, system monitors, camera, etc.; and ensure integrity and authenticity of digital evidence. The collected information is helpful in validating an incident.

Data Analysis Phase

Data analysis is another important phase where the collected data are analyzed to confirm the suspicious activities. It includes activities such as data extraction, event construction, data transformation into the most manageable size and form analysis, conducting an initial data survey to identify the digital evidence, and assessing the skill level of a suspect. It uses data extraction techniques such as keyword search, extraction of unallocated space and file slack, hidden data discovery, etc. The correlated events are analyzed, and the response is provided to answer the investigation questions.

Presentation of Findings Phase

The aim of this phase is to communicate the searching to the variety of audiences, technical personnel, legal staff, and law enforcement. The presentation provides the detailed event reconstruction information examined in the data analysis phase. The information is available in written and oral form helpful to those who then act upon it.

Incident Closure Phase

The phase focuses on the closure of the investigation. Various activities carried out at this stage are: conducting a critical review of the entire process; investigating to identify and apply lessons learned; disposing of evidence such as returning to the owner and destroying; cleansing, reusing, collecting and preserving all information related to the incidents. The knowledge gained from this phase is used to enhance subsequent investigations.

Second-Tier Phases (Subphases) The second tier consists of objective-based subphases task hierarchies subordinate to specify objectives of interest. The objective-based subphases (OBSP) remain consistent from situation to situation. Specific objective-based tasks are selected in each case. Useful matrices are used to form the tasks associated with each subphase. It enables the forensic examiner to determine which objective and which task apply to a particular incident, and they make the strategy accordingly. Investigators require an efficient mechanism to identify the tasks which need for the investigation at hand. Various digital investigation objectives are to determine if unauthorized system modifications have occurred and determine which accounts have been compromised. The proposed framework provides the development of objective task matrices to reduce cognitive burden. The abstraction layer is to analyze and transform data into a manageable and human-readable format. Layers of abstraction include physical media, media management, the file system application, and the network.

2.3 Network Forensic Process Models

Network forensics is a subfield of digital forensics related to the monitoring and analysis of computer network traffic for information gathering and legal evidence. An attacker might compromise the host, the network, or any device on the network and, hence, can modify or erase log files. Network forensics performs the investigation using an appropriate process model. Network tools such as TCPDump, Wireshark, PADS, Sebek, LiLK, TCPReplay, Snort, Bro, etc., are used in the investigation process. Various network forensic process models are explained below:

Ó Ciardhuáin [14] proposed an extended model of cybercrime investigations by combining existing models. This model represents the information flows and captures the full investigation. Awareness is the first step which announces an investigation. Authorization is granted by internal and external entities. Planning involves strategies and policies. Notification refers to informing the concerned parties about investigation. Search and identify evidences involve finding out the all possible sources of evidences such as computer, hard disk, server logs, etc. Collection of evidences is the activity in which evidences are collected from the sources identified. They are preserved and analyzed. Transport refers to transport of evidence to secure and suitable location. They are stored and examined for any damage or inaccuracy. Hypothesis is formulated and presented. For guiding future investigations and procedures, dissemination is performed.

Merkle [15] investigated the automated analysis of network-based evidence in response to cyberspace attacks. The complexity problem of analyzing raw traffic data and the quantity problem of the amount of data to analyze are two major challenges of network forensics which are addressed in his solution. The model is used to integrate results of data logged by various tools into a single system that can exploit computational intelligence to reduce human intervention. This integrated tool is referred as the automated network forensic tool. There are many stages in the network forensic analysis. An isolated network of virtual machines built into a honeynet. Open-source forensic tools are used for collecting the data. The information produced by various tools in one stage is characterized and transformed for use by other tools in the succeeding stages. Time-consuming and error-prone processes are identified and automated. The data sets are partitioned, and the system is trained and then tested.

Hu et al. [16] presented the blueprint and implementation of DDCFS: a distributed dynamic computer forensic system based on network. Evidence acquisition model, evidences transmission model, evidences storage model, evidences analysis model, and console model are described in context of DDCFS. These models are included in DDCFS proposed architecture. DDCFS helps to acquire potential electronic evidences. The process model is shown in Fig. 2.14.

Ming et al. [17] have proposed a new data collection model which collects the network data of a targeted system as shown in Fig. 2.15. This data is later used to detect the invasion on the fly by offline forensic analysis of the data. The result

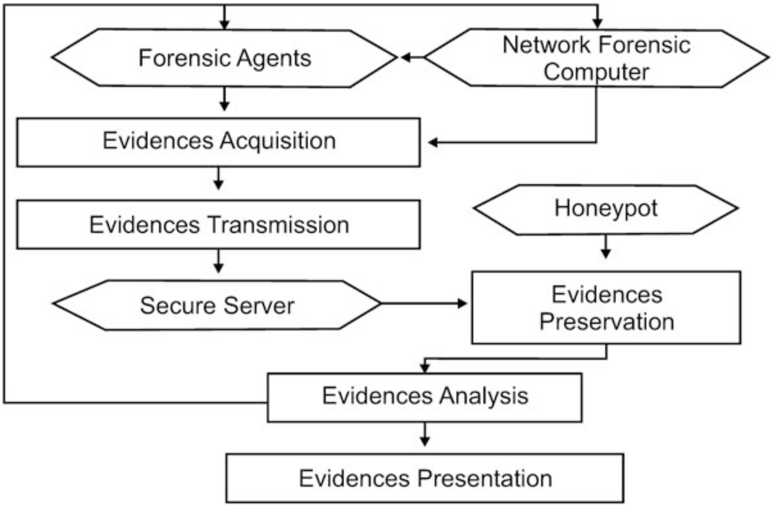


Fig. 2.14 Forensic procedure of DDCFS

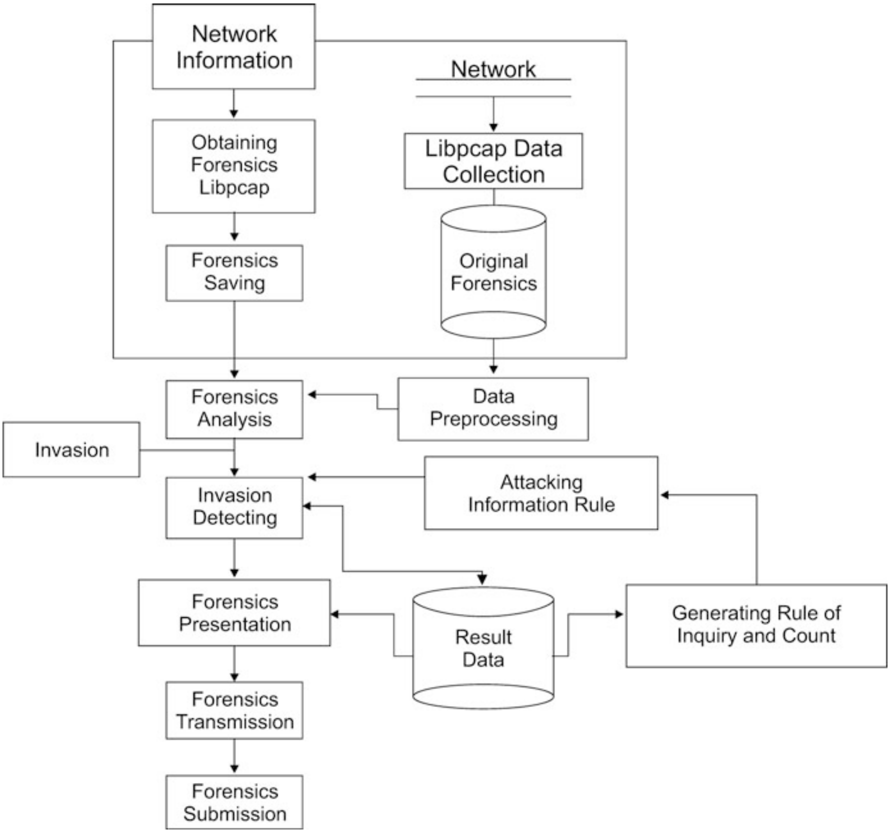


Fig. 2.15 Network forensic model

of forensic analysis takes the form of "rules," which helps in online detection of the invasion attempts. The paper further elaborates the characteristics of the model, realization problem of such model, preprocessing and analysis of the collected data.

Yong-Dal [18] presented a new digital forensic investigation procedure model which comprises the following phases: investigation preparation, classifying cyber-crime, deciding investigation priority, investigating damaged (victim) digital crime scene, criminal profiling consultant and analysis, tracking suspects, investigating injurer digital crime scene, summoning suspect, additional investigation, writing criminal profiling, and writing report as shown in Fig. 2.16. The investigation process starts with the preparation in which training and equipment are provided for the personnel. Initial report is provided to lab analyst, responders, and staff. Next is classifying the crime as violence or nonviolence. Investigators collect the damaged digital evidences and listen to testimony from managers. Damaged scenes must be documented and photographed. External experts can be invited to

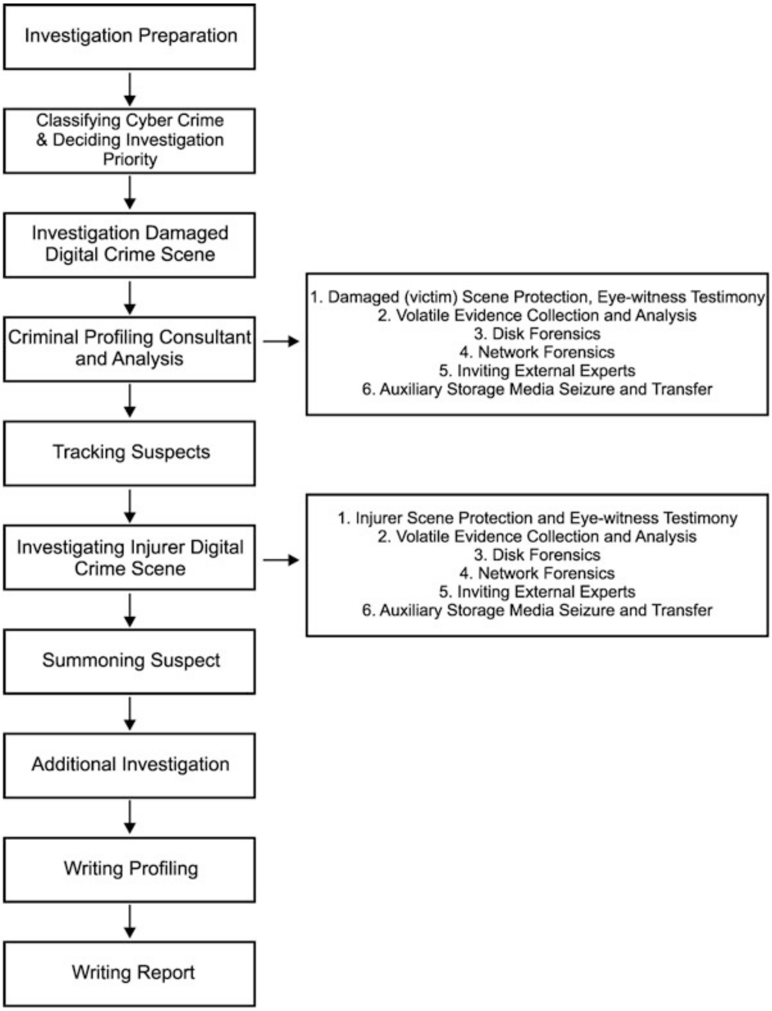

Investigation Preparation
Classifying Cyber Crime & Deciding Investigation Priority
Investigation Damaged Digital Crime Scene
Criminal Profiling Consultant and Analysis
Tracking Suspects
Investigating Injurer Digital Crime Scene
Summoning Suspect
Additional Investigation
Writing Profiling
Writing Report
1. Damaged (victim) Scene Protection, Eye-witness Testimony
2. Volatile Evidence Collection and Analysis
3. Disk Forensics
4. Network Forensics
5. Inviting External Experts
6. Auxiliary Storage Media Seizure and Transfer
1. Injurer Scene Protection and Eye-witness Testimony
2. Volatile Evidence Collection and Analysis
3. Disk Forensics
4. Network Forensics
5. Inviting External Experts
6. Auxiliary Storage Media Seizure and Transfer

Fig. 2.16 Digital forensic investigation procedure model

jointly perform investigation. Criminal profile is prepared which provides clues to cybercrime. Investigators trace the suspect and injurer criminal scene on game ID, IP address, MAC address, etc., obtained from damaged crime scenes. The next phase is investigating the injurer of the digital crime scene. Based on the evidences collected, investigators summon the suspect. Investigators decide to perform further investigation or stop here based on the findings of investigations. At last all findings are documented in form of a report to present the case to the court. The report should be easy to understand.

The proposed architecture of Strauss [19] consists of various phases. The initial phase is preparation. Here relevant forensic tools are identified. The forensic system should have a forensic agent with sufficient permissions to log the state of IPS tasks. Internal and external functions are used to detect the intrusion detection. Processes that violate the system policies should trigger the alarm and then record the data. Incidents are identified. Based on the nature of incidents, responses can vary. Resource allocation policies are dynamically adjusted to overcome the effect of suspicious behavior. Logical isolation in Recursive InterNetwork Architecture (RINA) can be achieved by inserting special forensic Distributed IPC Facility (DIF) in appropriate position. The traffic is logged in more detail. Forensic tools (sensors) are used to store data. In RINA, transmitted traffic and the binding between processes are stored. Bindings between port IDS, addresses and application names need to be recorded by agents. Only connection IDS are not sufficient. The collected data is preserved for further analysis. Next phase is examination, analysis, and investigation. The recorded data are thoroughly examined and analyzed. Only network traffic does not provide enough context to understand the header. Mapping of applications is done by examining logs of IPC processes.

Zainudin et al. [20] have presented a model of digital forensic investigation for online social networks. The model is proposed in two environments, physical and digital. At first, preliminary activities such as acknowledgment, construction, notification, and survey are discussed; investigation processes (identification, searching, filtering, and capturing) are discussed in detail. At last, analysis and evaluation (presentation, justification, and review) are performed. Some functionalities of application prototype are also discussed such as the autogeneration for data, ability to search and filter data, comprehensive report, time-efficient prototype, and ability to run and perform multiple searches. Application prototype is discussed in two modules: investigation of online social networks and analysis and generation of report. The architecture of the proposed model is shown in Fig. 2.17.

Network forensics faces many problems such as difficulty of data collection and analysis caused by mass data, the lack of standards processes of evidence collection, the poor network safety consciousness, etc. To solve some problems, Liu et al. [21] present a network forensic solution based on intrusion detection analysis which can record network intrusion behavior and analyze network data as shown in Fig. 2.18. It adopts dynamic and static methods to analyze network intrusion data and make detailed records of the data and log.

Chen [22] proposed a model to detect the stealthy self-propagating attacks. The audit traffic includes both the normal traffic and anomalous traffic from where evidence can be collected to find out the main cause of the attack. The proposed technique filters out the attack-irrelevant data and applies random moonwalk algorithm (RMW) to the rest of the data for further analysis; refer to Fig. 2.19. The network forensic mechanism consists of three phases, training, logging, and investigation. In the training phase, traces from historical data are collected and used as an input to build the normal behavior profile. This normal behavior becomes the input to the logging phase. Real-time traffic is given as input to the data reduction phase or logging phase. A normal traffic filter returns the learned behavior, based on that anomaly score is calculated. The anomaly score decides whether the given connection is normal or not. Those connections where the anomaly score is higher

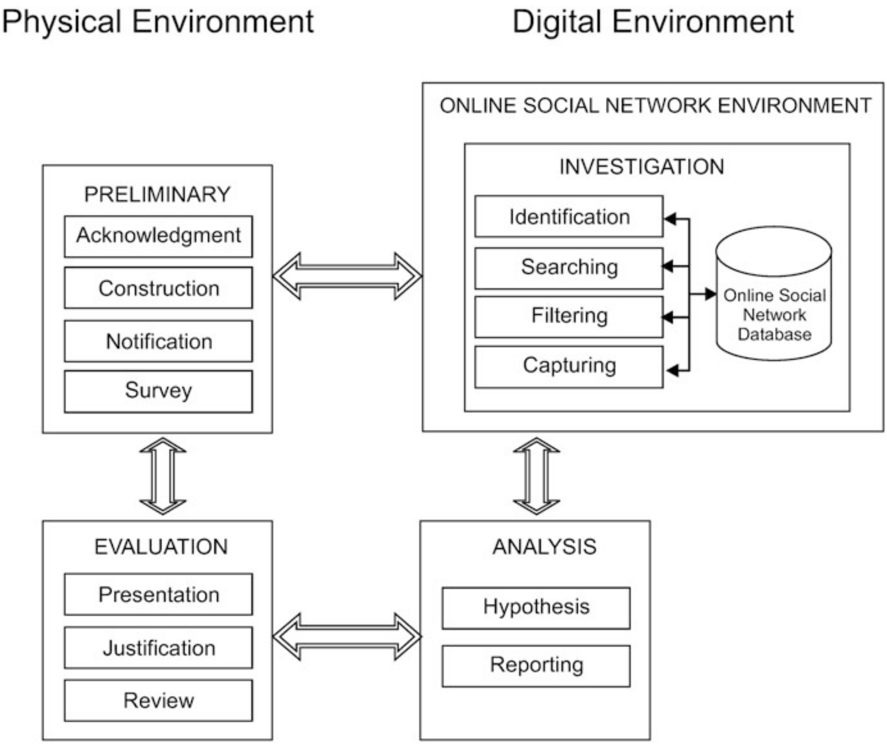


Fig. 2.17 Digital forensic model

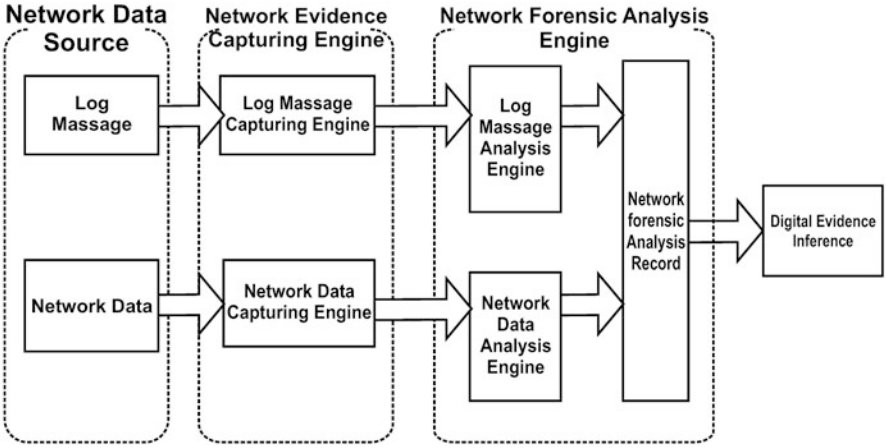

Network Data Source
Log Massage
Network Data
Network Evidence Capturing Engine
Log Massage Capturing Engine
Network Data Capturing Engine
Network Forensic Analysis Engine
Log Massage Analysis Engine
Network Data Analysis Engine
Network forensic Analysis Record
Digital Evidence Inference

Fig. 2.18 Network forensic system architecture

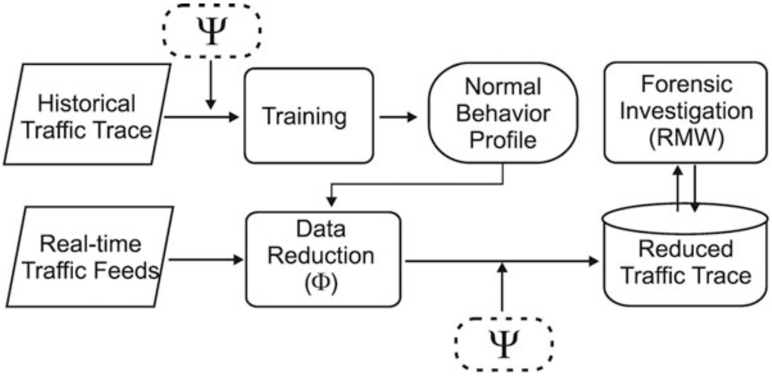


Fig. 2.19 Scalable network forensic mechanism

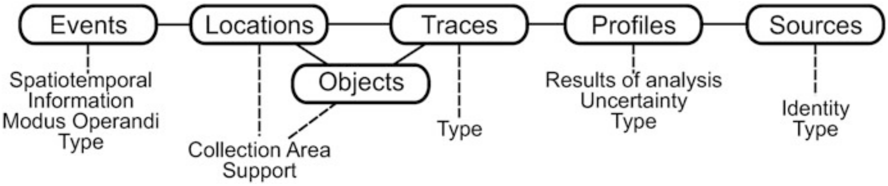


Fig. 2.20 Analysis procedure

than a particular threshold value are treated as unexpected and stored in reduced traffic set for further analysis. At the final phase RMW algorithm is applied to the reduced traffic trace to find out the type of attack and its origin. They used the learning-based data reduction and therefore separated collected real-time traffic into two parts: training and testing. A noise filter is used as a binary classifier for sanitizing the training data and the reduced traffic trace. Some rules can be applied to the noise filter to filter out some unwanted network activities.

Rossy et al. [23] have presented an approach using general intelligence process, analysis of investigation process, and visualization methods for forensic purpose as shown in Fig. 2.20. The rationality of the intelligence process and visualization information with its limitations are also discussed. Analysis of investigation problem through dominant dimensions are also shown, and these dimensions are quantitative, relational, spatial, and temporal. To design a link chart, a general guideline is suggested which includes defining the aim of the visualization by identification, identifying the relevant entities, handling the complexity of the problem, and being conscious about inappropriate visual choices. A theoretical model is also presented which combines the main entities such as events, locations, objects, profile, sources, and traces that are involved to send information to laboratory for forensic analysis.

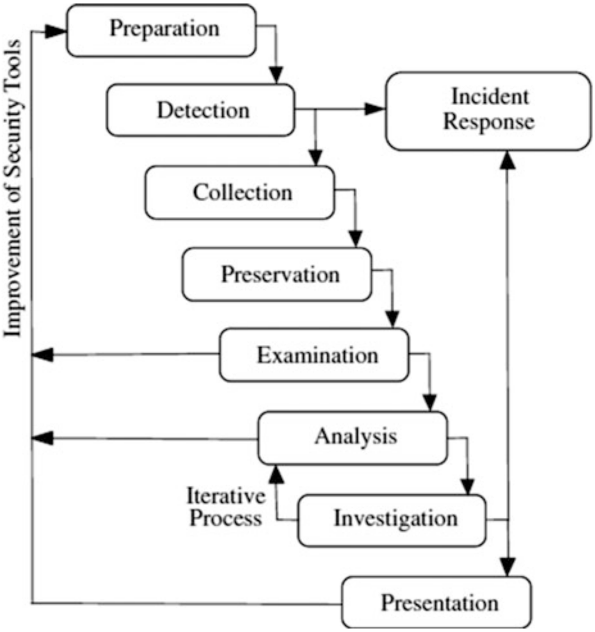
2.4 Generic Process Model for Network Forensics

Pilli et al. [24] has proposed a forensic network framework based on existing digital forensic models; refer to Fig. 2.21. An overview of about thirty-three network forensic analysis tools is also presented. An exhaustive survey was done for forensic network frameworks in six categories which are distributed system-based frameworks, soft computing-based frameworks, honeypot-based frameworks, attack graph-based frameworks, formal method-based frameworks, and aggregation frameworks. Different phases of the proposed network forensic framework are preparation, detection, incident response, collection, preservation, examination, analysis, investigation, and presentation. Research challenges are identified in the following sections which are collection and detection, data fusion and examination, analysis, investigation, and incident response. Various phases are explained below.

Preparation

In this phase various security tools such as intrusion detection system, firewall, packet analyzer, and traffic flow measure measurement are deployed at various strategic points on the network. All necessary legal documents are also acquired so that privacy is not violated. A proper training must be given to the staff working in this phase in order to facilitate attribution of the crime. The preparation phase reduces the overall cost of investigation.

Fig. 2.21 Generic process model for network forensics



Detection

This phase is related to generating alerts for detected threats. The anomalies and illegitimate events are analyzed based on various parameters, and attacks are detected. By a quick validation the suspected attack is confirmed. Accordingly an important decision is taken whether to continue the investigation and generate the alert or ignore the alert. Various tools are used such as TcpDump, Wireshark, PADS, Sebek, Ntop, p0f, Bro, Snort, etc. This phase branches in two directions, incident response and collection.

Incident Response

The response to the crime is dependent on the gathered information to validate and assess the incidents. An organization policy is kept in place while responding to attack. The response also depends on the type of the attack identified. Meanwhile, a decision is taken whether to continue with the investigation or gather more information. An important criterion to respond to an incident while performing network forensic analysis is to ensure that data being collected as evidence is neither tampered nor obstructed.

Collection

Data are acquired from the sensors used to collect the traffic data. The sensors used must be secure, must be fault tolerant, must have limited access, and must be able to avoid compromise. A well-defined procedure using reliable hardware and software tools must be in place to gather maximum evidence causing minimum impact to the victim. The network must be monitored to identify future attacks. The integrity of data logged and network events recorded must be ensured. This phase is very significant as the traffic data change at a rapid pace, and it is not possible to generate the same trace at a later time. Due to the enormous amount of data logged, huge memory space is required, and the system must be versatile in nature and able to handle different log data formats appropriately. Various tools used in this phase are TCPDump, Wireshark, PADS, Sebek, LiLK, TCPReplay, Snort, Bro, etc.

Preservation

The collected traces and logs are stored on a backup device like read-only media. A hash of all the traces is preserved. The analysis is carried out over the copied data, and the original data is kept untouched. This step is carried out to prove the investigation when the process is repeated on original data to facilitate the legal requirements. Various tools used in this phase are TCPDump, Wireshark, PADS, Sebek, LiLK, TCPReplay, Snort, Bro, etc.

Examination

Traces are integrated and fused as a large data set on which the analysis is performed. There can be some issues like redundant information and overlapping time zones which need appropriation. Alerts from various sources may be contradictory. However, this process needs to be done in such a manner so that crucial information from important sources is not lost. The collected data is classified and clustered into groups so that the volume of data to be stored may be reduced to manageable chunks. It is easy to analyze large groups of organized data. The collected evidence is searched methodically to extract specific indicators of the crime. Minimum attack attributes selected must be so credible that the least information recorded, contains the highest probable evidence. Feedback is given to improve the security tools. Various tools used are TCPDump, Wireshark, TCPFlow, Flow-tools, NfDump, PADS, Argus, Nessus, Sebek, TCPTrace, Ntop, TCPStat, NetFlow, TCPDstat, ngrep, TCPXtract, SiLK, TCPReplay, P0f, Nmap, Bro, Snort, etc.

Analysis

This phase performs analysis on data using various approaches like data mining such as ANN, fuzzy, and genetic algorithm (GA) and statistical computing to search of attack patterns in the data. Some of the critical parameters are related to network connection establishment, DNS queries, packet fragmentation, protocol, and operating system fingerprinting. The attack patterns are reconstructed and replayed to understand the intention and methodology of the attacker. Feedback is given to improve the security tools. The tools include TCPDump, Wireshark, TCPFlow, Flow-tools, NfDump, PADS, Argus, Nessus, Sebek, TCPTrace, Ntop, TCPStat, NetFlow, TCPDstat, ngrep, TCPXtract, SiLK, TCPReplay, P0f, Nmap, Bro, Snort, etc.

Investigation

Investigation is performed to determine the path from a victim network or system through any intermediate systems and communication pathways back to the point of attack origination. Packet statistics are obtained for attribution of the attack. This phase may require some additional features from the analysis phase, and hence these two phases are iteratively performed to arrive at the conclusion. Attribution is establishing the identity of the attacker and is the most difficult of network forensics. The investigation phase provides data for incident response and prosecution of the attacker.

Presentation

The observations are presented in an understandable language for legal personnel while providing an explanation of the various procedures used to arrive at the conclusion. Legal requirements must be fulfilled, and systematic documentation is presented to authorities. The findings are also presented using visualization so that they can be easily grasped. The statistical data is interpreted in support of the conclusions arrived. A thorough review of the incident is done, and countermeasures are recommended to prevent similar incidents in the future. The entire case is documented to influence future investigations and to provide feedback to guide the deployment and improvement of security products. This process concludes the network forensic analysis as the information presented results in the prosecution of the attacker.

2.5 Conclusion

The digital forensic process models proposed the last 15 years have been described in this chapter. The chronological development of the various phases is also brought out. The forensic standpoint from various dimensions is explored. Process models specific to network forensics are discussed while differentiating them with the computer forensic models. Additional features and aspects to be considered while analyzing network traffic are also highlighted. In conclusion a generic process model for network forensics is discussed. It is generic as it handles network forensics, both in real-time and post attack scenarios. The first few phases handle real-time network traffic and the remaining phases are common for real-time and post attack scenarios.

2.6 Questions

Multiple Choice Questions

Select the most suitable answer for the following questions:

1. What is the full form of DFRWS?

   (a) Digital Forensic Research Workshop
   (b) Digital Forensic Research Workgroup

   (c) Digital Forensics and Research Work
   (d) none of above

2. A network sniffer program is an example of:

   (a) Evidence development tool
   (b) Packet collection tool
   (c) Packet formatting tool
   (d) none of the above

3. Items included in a forensic toolkit should include the following except

   (a) Screwdrivers
   (b) Power cables
   (c) Printer
   (d) Permanent markers

4. The evidence custodian should

   (a) Give the evidence to the secretary
   (b) Place evidence in the storage place
   (c) Keep logs of who has the evidence, when was it check out, etc.
   (d) Use the evidence for personal use.

5. ______________ is forensics applied to information stored or transported on network.

   (a) Information forensics
   (b) Data forensics
   (c) Computer forensics
   (d) Network forensics

6. In ____________ intrusion detection system is a device or application used to inspect all network traffic and alert the user or administrator when there has been unauthorized attempts or access.

   (a) Alert data
   (b) Security check
   (c) Network security
   (d) Traffic control

7. Which phase is not included in generic process model?

   (a) Validation and Discrimination
   (b) Collection
   (c) Analysis
   (d) Investigation

8. What is full form RMW algorithm?

   (a) Ready and moonwalk
   (b) Ready modify and write

(c) Random moonwalk
(d) none of the above

9. CDE refers to

(a) Comprehensive Digital Evidence
(b) Common Digital Evidence
(c) Common Data Entity
(d) none of the above

10. MAC refers to

(a) Medium Access Control
(b) Machine Address Control
(c) Machine Assess Control
(d) none of the above

Short-Answer Questions

Write the brief answers of the following questions.

1. What is the difference between digital forensics and network forensics?
2. What is the role of sensors in digital investigation process?
3. Write any three techniques used for carrying out investigation?
4. What is first network forensic model?
5. What activities are carried out in preservation phase?
6. Why do private networks can be a richer source of evidence than the Internet ?
7. What is the goal of an investigation?
8. What is hierarchical process model?
9. Name some tools used in carrying out the investigation.
10. What are sources from where information is collected?

Long-Answer Questions

Write in detail the answer of following questions:

1. An organization is using its own servers, network, and other networking resources. Data travels form host to hosts via switches and routers. What are various places from where data can be collected and in which form? Which process model is suitable for this type of investigation if a criminal activity is detected. Explain the model.
2. What information can you gain by capturing and analyzing cache files? How might this prove useful in the court of law?
3. What are the various phases in multi-tier and hierarchical process model. Explain with a dia.
4. A generic process model is different from other process models. What are the phases and tools used by it? Write sufficient measures to improve the investigation process incorporated by this model.
5. Write a detailed note on sources of investigation and tools for data collection in digital investigation process model.

References

1. Palmer G (2001) A road map for digital forensic research. Utica, New York
2. Carrier B, Spafford EH (2003) Getting physical with the digital investigation process. Int J Digit Evid 2(2):1–20
3. Baryamureeba V, Tushabe F (2004) The enhanced digital investigation process model. In: Fourth Digital Forensic Research workshop, pp 1–9
4. Casey E, Palmer G (2004) The investigative process. In: Digital evidence and computer crime. Elsevier Academic Press, London
5. Ieong RSC (2006) FORZA-Digital forensics investigation framework that incorporate legal issues. Digit Investig 3:29–36
6. Selamat SR, Yusof R, Sahib S (2008) Mapping process of digital forensic investigation framework. Int J Comput Sci Netw Secur 8(10):163–169
7. Grobler CP, Louwrens CP, von Solms SH (2010) A multi-component view of digital forensics. In: ARES'10 international conference on availability, reliability, and security, pp 647–652
8. Ademu IO, Imafidon CO, Preston DS (2011) A new approach of digital forensic model for digital forensic investigation. Int J Adv Comput Sci Appl 2(12):175–178
9. Agarwal A, Gupta M, Gupta S, Gupta SC (2011) Systematic digital forensic investigation model. Int J Comput Sci Secur (IJCSS) 5(1):118–131
10. James JI, Gladyshev P (2013) A survey of digital forensic investigator decision processes and measurement of decisions based on enhanced preview. Digit Investig 10(2):148–157
11. Shrivastava AK, Payal N, Rastogi A, Tiwari A (2013) Digital forensic investigation development model. In: 5th international conference on Computational Intelligence and Communication Networks (CICN), pp 532–535
12. Kohn MD, Eloff MM, Eloff JHP (2013) Integrated digital forensic process model. Comput Secur 38(1):103–115
13. Beebe NL, Clark JG (2005) A hierarchical, objectives-based framework for the digital investigations process. Digit Investig 2(2):147–167
14. Ciardhuain SO (2004) An extended model of cybercrime investigations. Int J Digit Evid 3(1):1–22
15. Merkle LD (2008) Automated network forensics. In: Genetic and Evolutionary Computation Conference (GECCO 08), Atlanta, Georgia, USA, pp 1929–1932
16. Liang H, Kuo T, Guangkun S, Nurbol, Kuo Z (2009) DDCFS: a distributed dynamic computer forensic system based on network. In: Second international conference on intelligent computation technology and automation, pp 53–56
17. Shin YD (2008) New digital forensics investigation procedure model. In: 2008 fourth international conference on networked computing and advanced information management, pp 528–531
18. Hou M, Shen L (2009) A new system design of network invasion forensics. In: Second International Conference on Computer and Electrical Engineering (ICCEE), Dubai, pp 596–599
19. Strauss T, Olivier MS (2011) Network forensics in a clean-slate Internet architecture. In: Information Security South Africa (ISSA), pp 1–5
20. Zainudin NM, Merabti M, Llewellyn-Jones D (2011) Online social networks as supporting evidence: a digital forensic investigation model and its application design. In: International Conference on Research and Innovation in Information Systems (ICRIIS), Kuala Lumpur, Malaysia, pp 1–6
21. Jiang L, Tian G, Zhu S (2012) Design and implementation of network forensic system based on intrusion detection analysis. In: International conference on Control Engineering and Communication Technology, pp 689–692
22. 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

23. Rossy Q, Ribaux O (2014) A collaborative approach for incorporating forensic case data into crime investigation using criminal intelligence analysis and visualisation. Sci Justice 54(2):146–153
24. Pilli ES, Joshi RC, Niyogi R (2010) Network forensic frameworks: survey and research challenges. Digit Investig 7(1–2):14–27

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

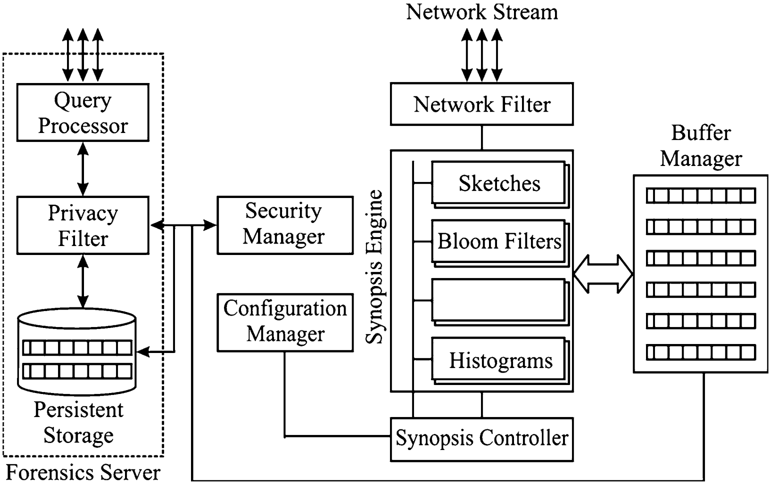


Fig. 3.1 Architecture of a SynApp with an integrated forensic server

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

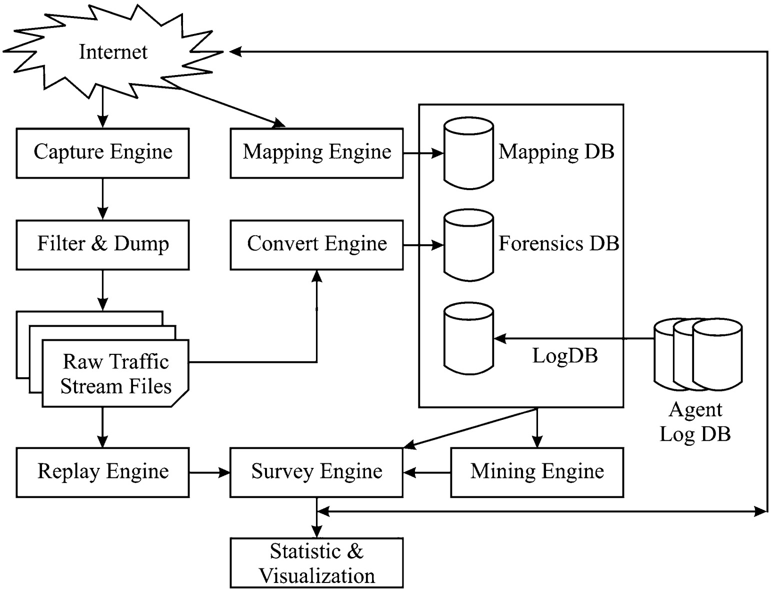


Fig. 3.2 The architecture of the network forensic server

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

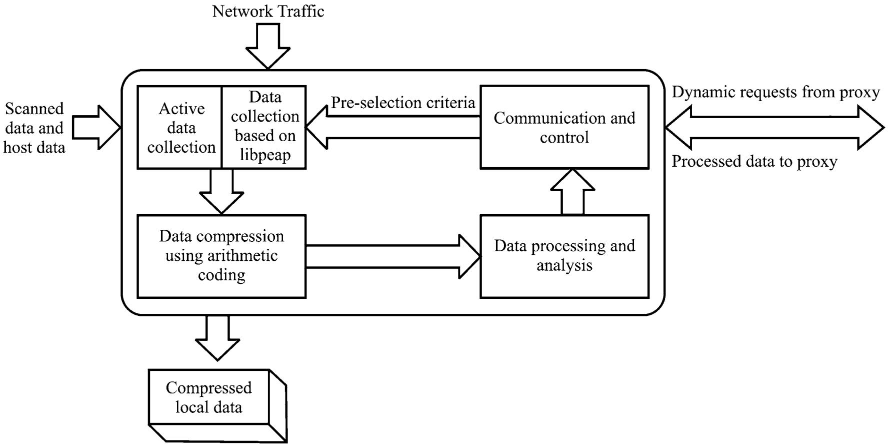

Network Traffic
Scanned data and host data
Active data collection
Data collection based on libpeap
Pre-selection criteria
Communication and control
Dynamic requests from proxy
Processed data to proxy
Data compression using arithmetic coding
Data processing and analysis
Compressed local data

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

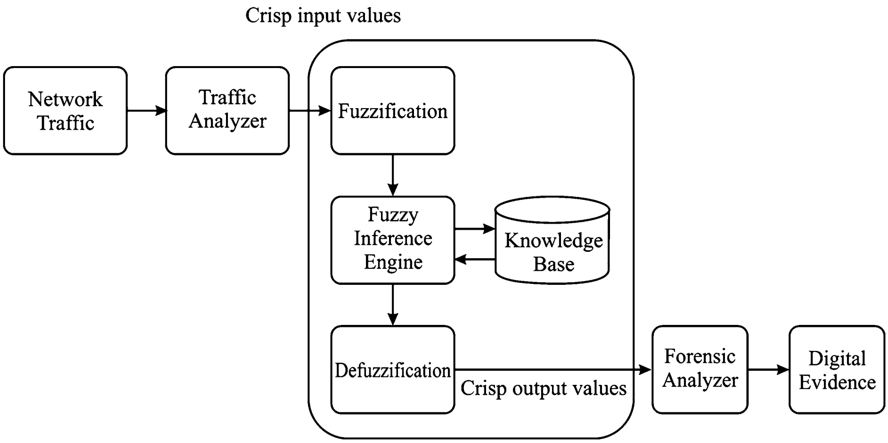


Fig. 3.4 The architecture of the fuzzy expert system for network forensics

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.

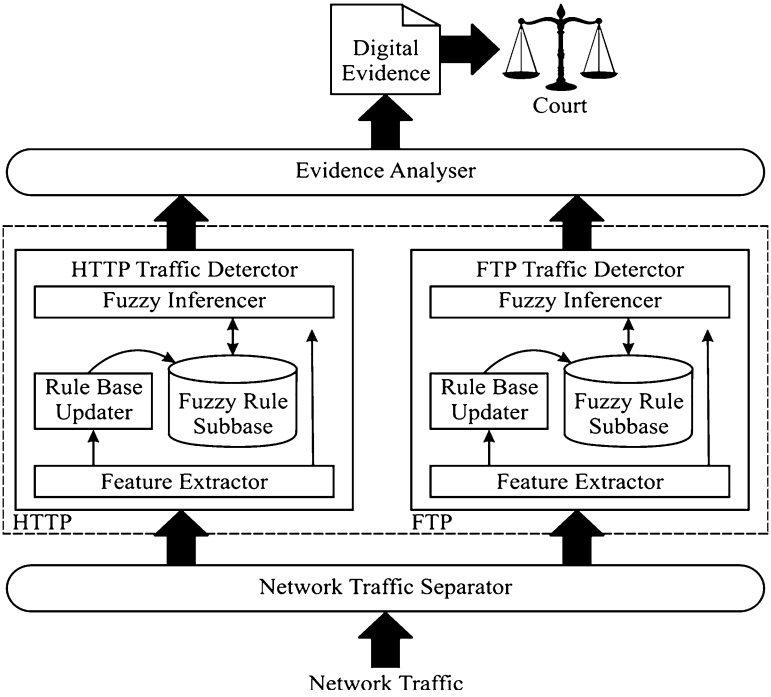


Fig. 3.5 Incremental fuzzy decision tree technology network forensic system

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.

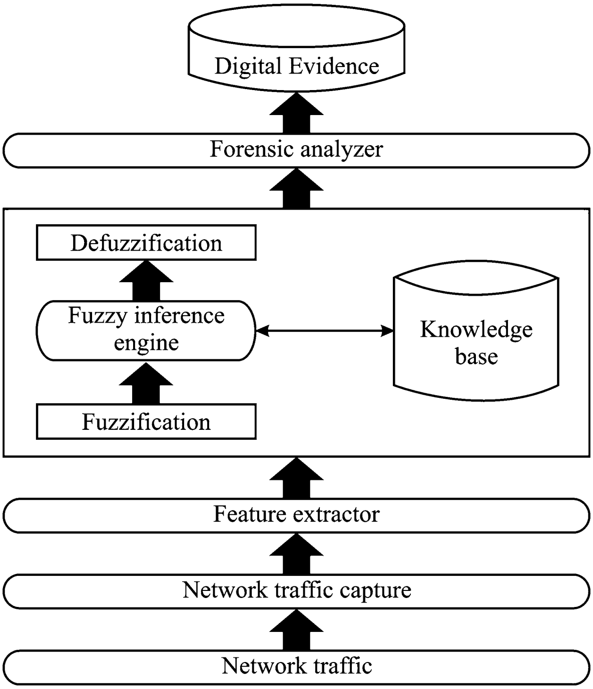


Fig. 3.6 The architecture of the proposed network forensic system

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

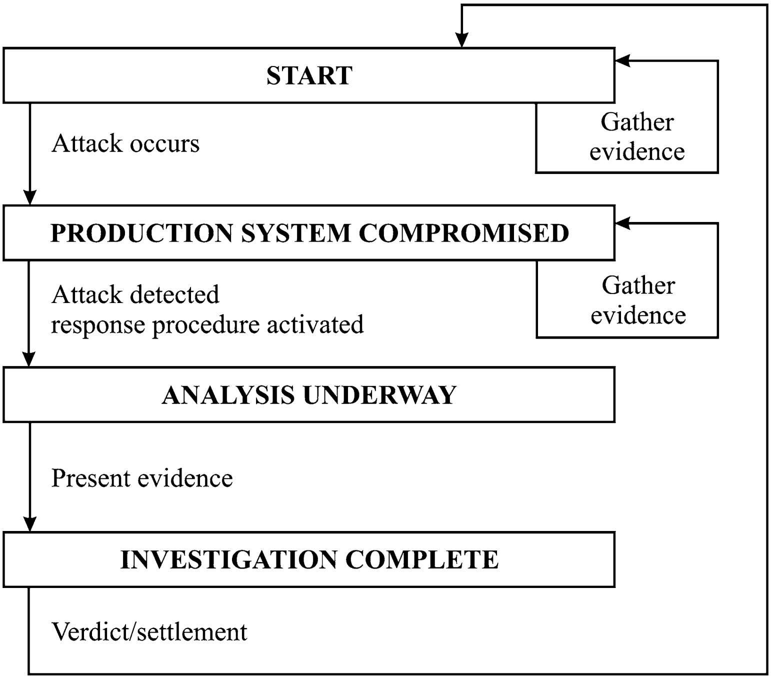


Fig. 3.7 Production system forensic investigation

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.

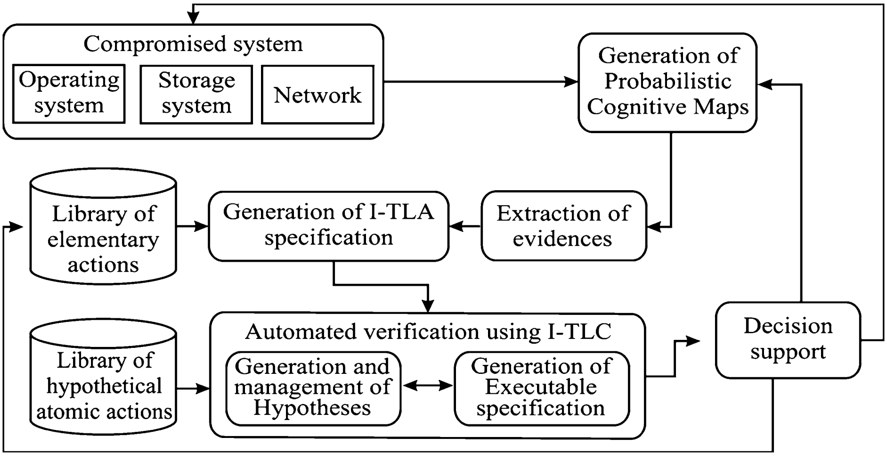

Compromised system
Operating system
Storage system
Network
Generation of Probabilistic Cognitive Maps
Library of elementary actions
Generation of I-TLA specification
Extraction of evidences
Library of hypothetical atomic actions
Automated verification using I-TLC
Generation and management of Hypotheses
Generation of Executable specification
Decision support

Fig. 3.8 DigForNet methodology

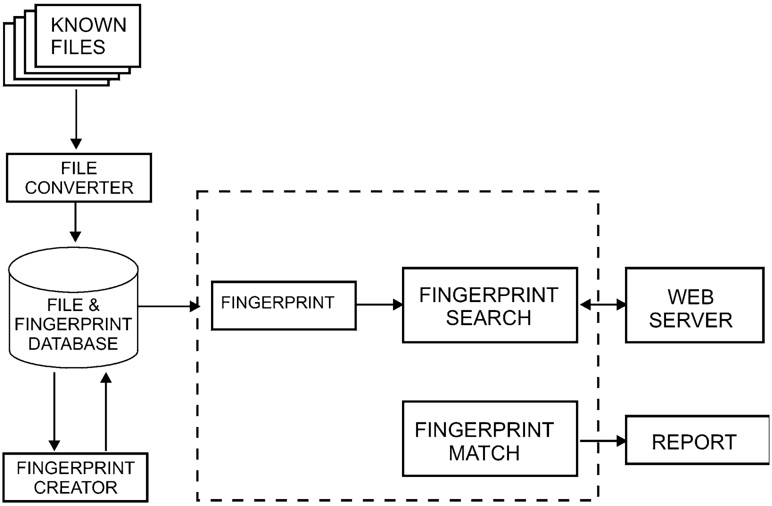


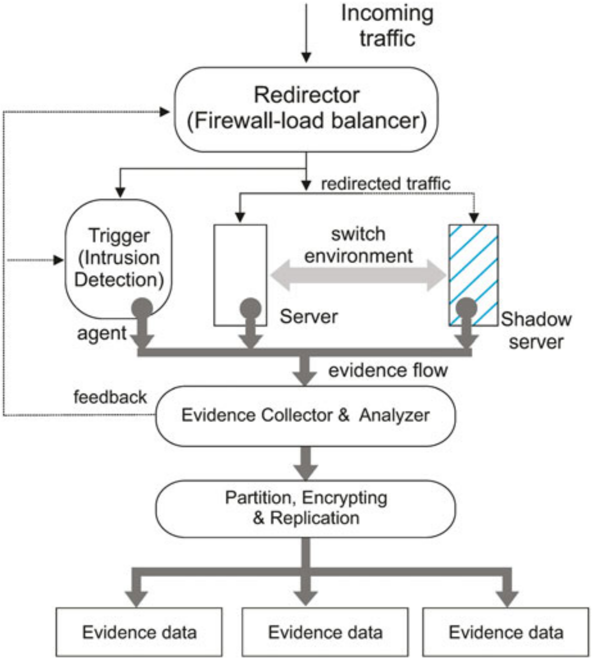
Fig. 3.9 Overview of the FORWEB application

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

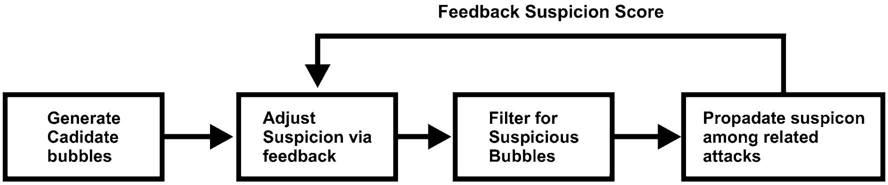

Feedback Suspicion Score
Generate Cadidate bubbles
Adjust Suspicion via feedback
Filter for Suspicious Bubbles
Propadate suspicon among related attacks

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

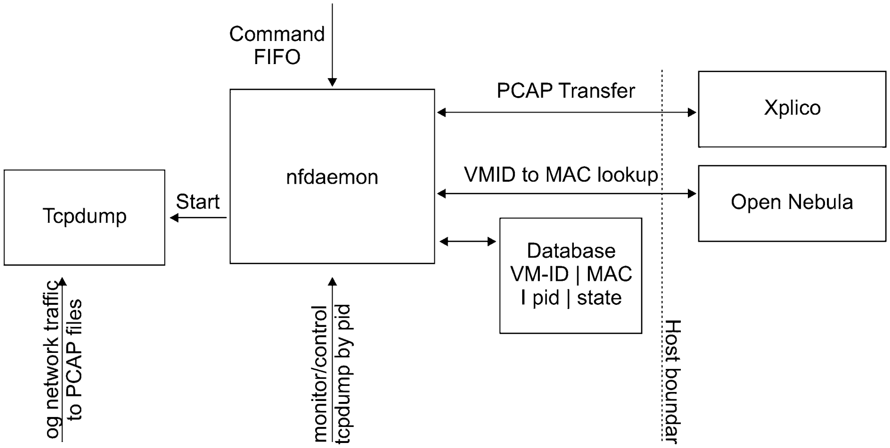


Fig. 3.12 Network forensic daemon and its components

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.

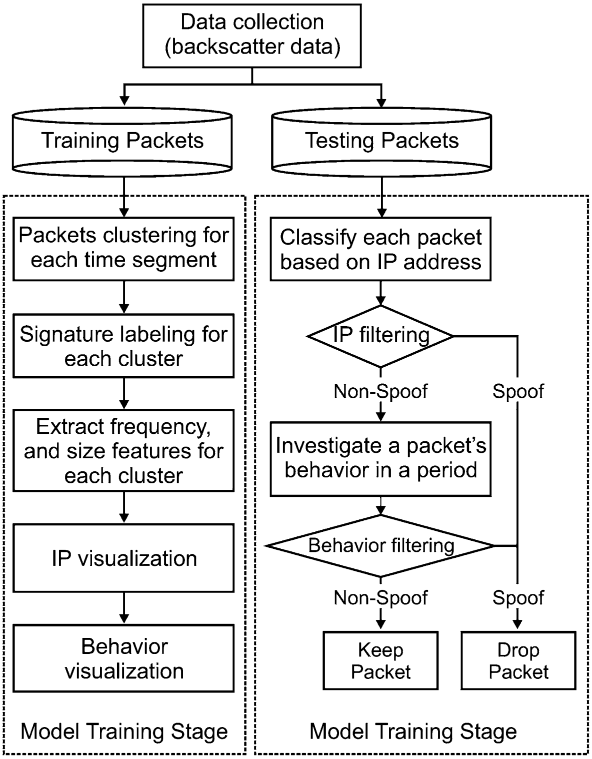

Data collection (backscatter data)
Training Packets
Testing Packets
Packets clustering for each time segment
Signature labeling for each cluster
Extract frequency, and size features for each cluster
IP visualization
Behavior visualization
Model Training Stage
Classify each packet based on IP address
IP filtering
Non-Spoof
Spoof
Investigate a packet's behavior in a period
Behavior filtering
Non-Spoof
Spoof
Keep Packet
Drop Packet
Model Training Stage

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

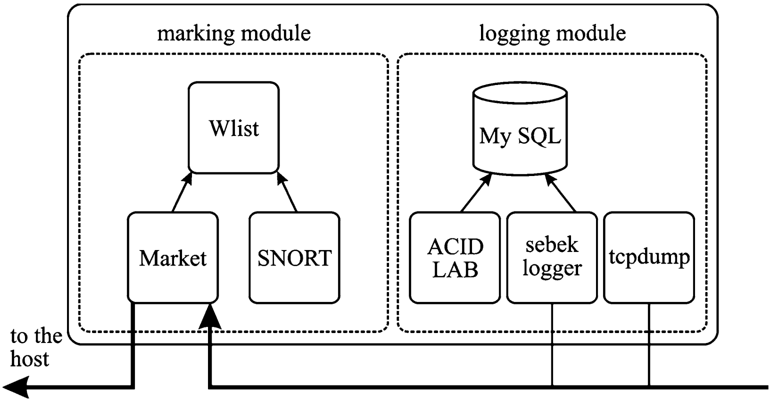


Fig. 3.14 The bridge internal

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.

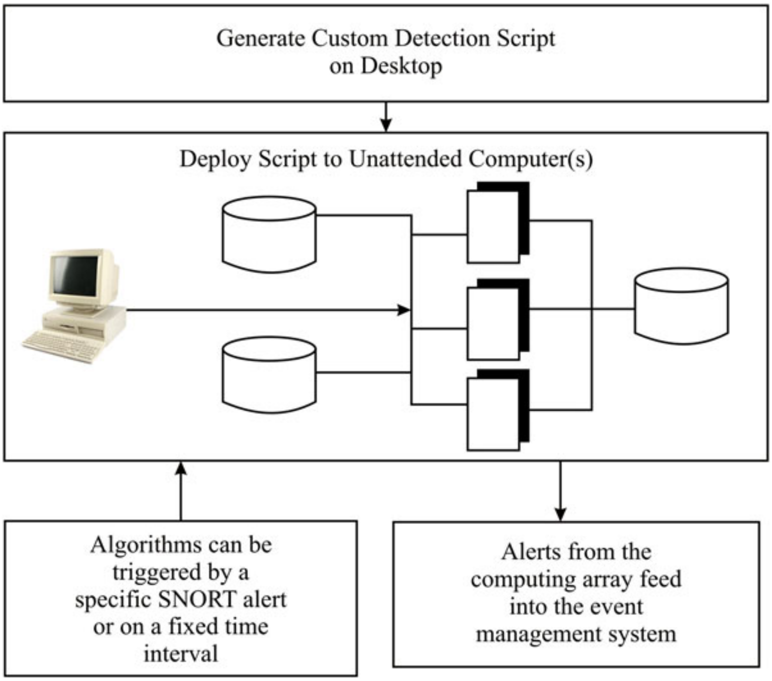

Generate Custom Detection Script on Desktop
Deploy Script to Unattended Computer(s)
Algorithms can be triggered by a specific SNORT alert or on a fixed time interval
Alerts from the computing array feed into the event management system

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.

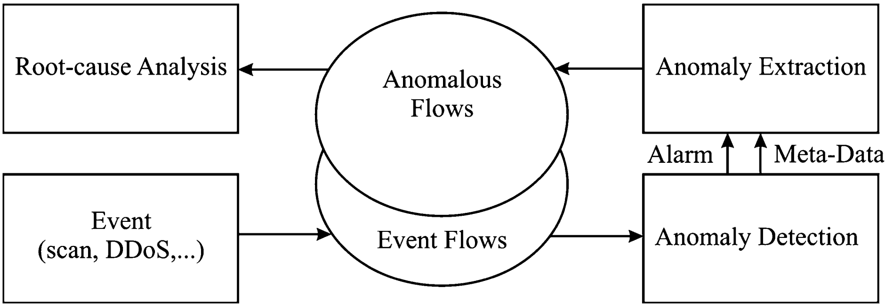


Fig. 3.16 High-level goal of anomaly extraction

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 multi-conference 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

Chapter 4
Network Forensic Tools

Learning Objectives

- Understand and use of the Network Forensic Analysis Tools (NFATs)
- Understand and use the Vulnerability Assessment, Scanning, Sniffing and Packet Analysis tools
- Study of some popular Network Security and Monitoring (NSM) tools

4.1 Introduction

Network forensic tools allow us to monitor networks, gather information about the traffic, and assist in network crime investigation. Forensic tools help in analyzing the insider theft, misuse of resources, predict attack targets, perform risk assessment, evaluate network performance, and protect intellectual propriety. Forensic tools can capture the entire network traffic, allow users to analyze network traffic according to their needs and discover significant features about the traffic. Forensic tools synergize with intrusion detection systems and firewalls and make long-term preservation of network traffic records for quick analysis. These tools are called network forensic analysis tools (NFAT).

There are some forensic tools available that provide reliable data acquisition and powerful analysis capabilities. There are many other open-source Network Security and Monitoring (NSM) tools, which were developed to provide network security. They were not designed with evidence gathering and processing in mind. However they can be used to help in specific activities of forensic analysis.

We have categorized the NSM tools in five distinct categories based upon their functionality, viz., vulnerability assessment tools, network sniffers and packet analysis tools, network scanning tools, network monitoring tools, and intrusion detection systems (IDS). The list of tools given is not exhaustive.

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_4

4.2 Network Forensic Analysis Tools (NFAT)

In this section we describe some of the popular network forensic analysis tools (NFATs). The tools are both proprietary and open source. Proprietary tools are built with an appliance for logging, recording and storing network data as evidence for evaluation. Logged data can be stored for more than a year to investigate the network events which are time based. Open source tools are mostly software, built for Linux environments. They can be easily reprogrammed to add more additional functionalities. They can also be mastered and can also be aggregated to become a new and powerful tool.

NetDetector

NetDetector [1] is a full-featured forensic tool built on NIKSUN's Alpine architecture. It integrates signature-based IDS with statistical anomaly detection with full-application reconstruction, packet level decoding, etc. NetDetector informs the user about the security breaches and can take preventive measures such as blocking the malicious traffic from entering the system.

Some key features of the NetDetector are as follows:

- Big data security and intelligence
- Reconstruction of applications and sessions
- Integrated signature-based IDS and anomaly detection
- Traffic capture and multi-timescale analysis
- Ad hoc and scheduled reporting

NetIntercept

NetIntercept [2] is available as a complete system, with hardware and pre-installed software, ready to deploy forensic solution. NetIntercept can be placed in the machine room or at the firewall. NetIntercept records the traffic in hard disk; hence, traffic from the last hours, days, or weeks is available for the analysis. The analysis is generally analyzed in batch mode.

NetIntercept correlates user sessions and reconstructs transmitted or received files, providing immediate proof of the misbehavior. NetIntercept aims to answer following basic questions:

- Who sent what information where
- Why information isn't moving
- How the system was attacked

OmniPeek

OmniPeek [3] provides real-time visibility into every part of the network. It has high capture capabilities, centralized console, distributed engines, and expert analysis. OmniPeek supports Ethernet, Gigabit, 10 Gigabit, 802.11a/b/g/n/ac wireless, VoIP, video, MPLS, and VLAN. OmniPeek is available in four versions: basic, professional, enterprise, and connect. Each version provides some unique features and supports different types of networks.

Python Forensic Log Analysis GUI (PyFLAG)

PyFLAG [4] is a Web-based, database-backed forensic and log analysis tool written in python. PyFlag analyzes captured packets in libpcap format while supporting a number of network protocols. It has the ability to recursively examine data at multiple levels and is ideally suited for network protocols which are typically layered. PyFlag parses the pcap files, extracts the packets, and dissects them at low-level protocols (IP, TCP, or UDP). Related packets are collected into streams using a reassembler. These streams are then dissected with higher-level protocol dissectors such as HTTP, IRC, etc.

Xplico

Xplico [5] is an open-source forensic analysis tool for UNIX-like systems. Xplico is capable of reconstructing a protocol's application data from captured packets. Xplico was specifically created for reconstruction of protocol data. It employs as technique named Port Independent Protocol Identification (PIPI) for recognizing the protocols. Xplico dissects data at the protocol level and reconstructs and normalizes it for use in manipulators. The manipulators then transcode, correlate, and aggregate data for analysis and present the results in a visualized form.

4.3 Vulnerability Assessment Tools

Vulnerabilities are regrettably an integral part of a computer-based system whether software or hardware. A bug in a commercial software or a loophole in the system, bugs in the operating system, misconfigurations, etc., make systems susceptible to malicious attacks or access. A malicious person can access the system using these loopholes and bugs. From a technical point of view, such an attempt is not easy, but there have been incidents in the past that have cost reputation and money. Known and unknown vulnerabilities can be exploited by malicious users as well as authenticated but unauthorized person both from within and outside the organization.

Vulnerability assessment tools aims at finding such vulnerabilities in a system. Assessment tools scan the system for known vulnerabilities and sometimes mask a fake attack to find new vulnerabilities. Some popular tools are given below.

Metasploit

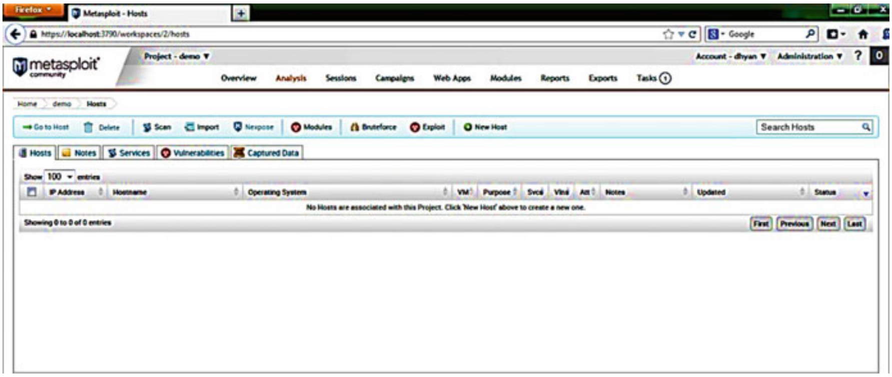
Metasploit [6] is basically a penetration testing framework. It provides a platform for developing, testing, and using "exploits" to take advantage of a system's vulnerabilities. Originally Metasploit was open source but Rapid7 acquired it in 2009. It is available for Windows, Linux, and MAC in GUI and a CLI-based interface.

Metasploit is available in three versions: Metasploit Community Edition which is free but has limited functionalities, Metasploit Express with some additional advanced features, and Metasploit Pro a full-featured version. Metasploit framework is still open source and available for download with tools for inspections and writing new exploits. Screenshot is shown in Fig. 4.1.

Nessus

Nessus [7] is a vulnerability-scanning tool for Windows, Linux, Solaris, FreeBSD, and MAC. Nessus is owned by Tenable Network Security. It is available in four versions, Nessus evaluation, Nessus, Nessus Perimeter Service, and Nessus Home.

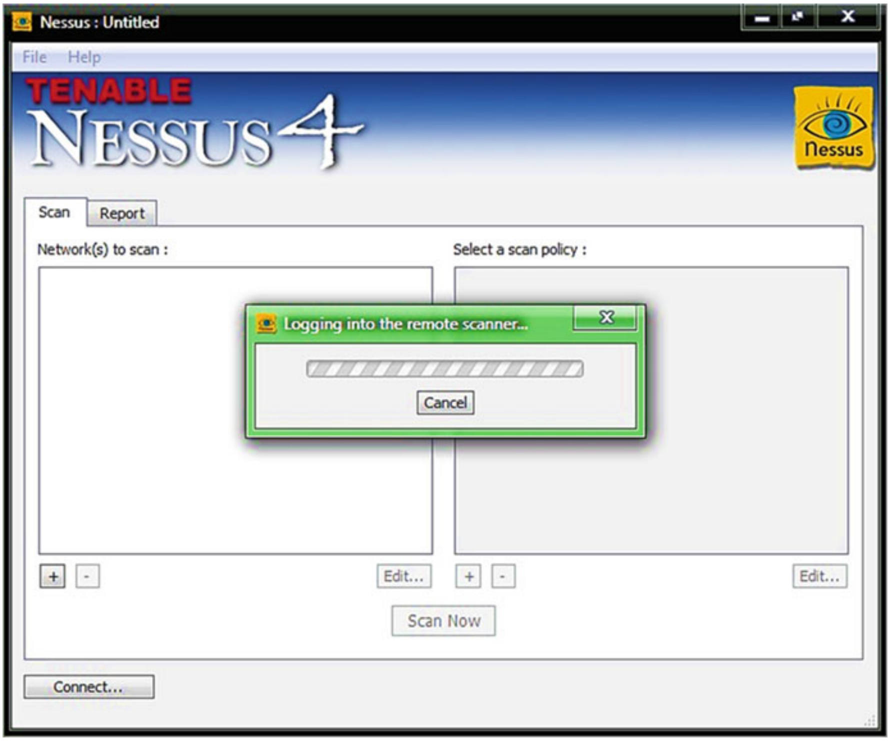
Nessus offers new vulnerability checks in a form of "plug-in" on a daily basis. It is also used for DoS attack scan, port scan, password vulnerabilities. The



□	IP Address	Hostname	Operating System	VM	Purpose	Svcs	Vlns	Att	Notes	Updated	Status
No Hosts are associated with this Project. Click 'New Host' above to create a new one.											

Showing 0 to 0 of 0 entries

Fig. 4.1 Metasploit: host analysis dashboard



Logging into the remote scanner...

Fig. 4.2 Nessus: logging in to server

vulnerability checks are available for free for home users and written in Nessus Attack Scripting Language (NASL). Screenshot is shown in Fig. 4.2.

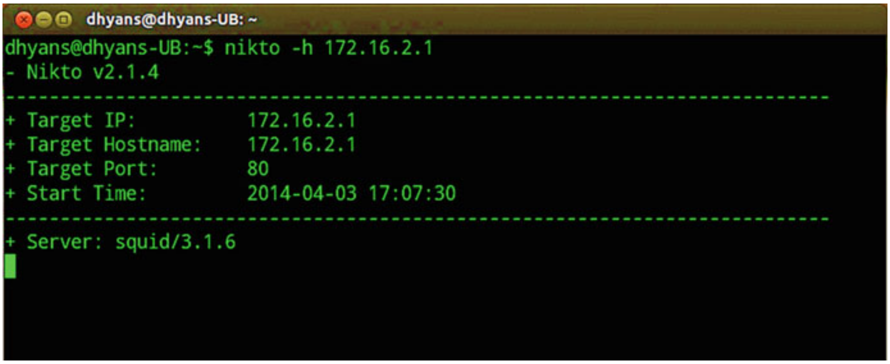
Nikto

Nikto [8] is an open-source (GPL) Web server vulnerability scanner. Nikto scans the server for dangerous files/CGIs, outdated server softwares, and version-specific bugs. It also checks for server configuration items such as index files, server options, etc. Screenshot is shown in Fig. 4.3.

Nikto presently scans for over 6700 dangerous files/CGIs, version-specific bugs on over 270 servers, and outdated version of over 1250 servers.

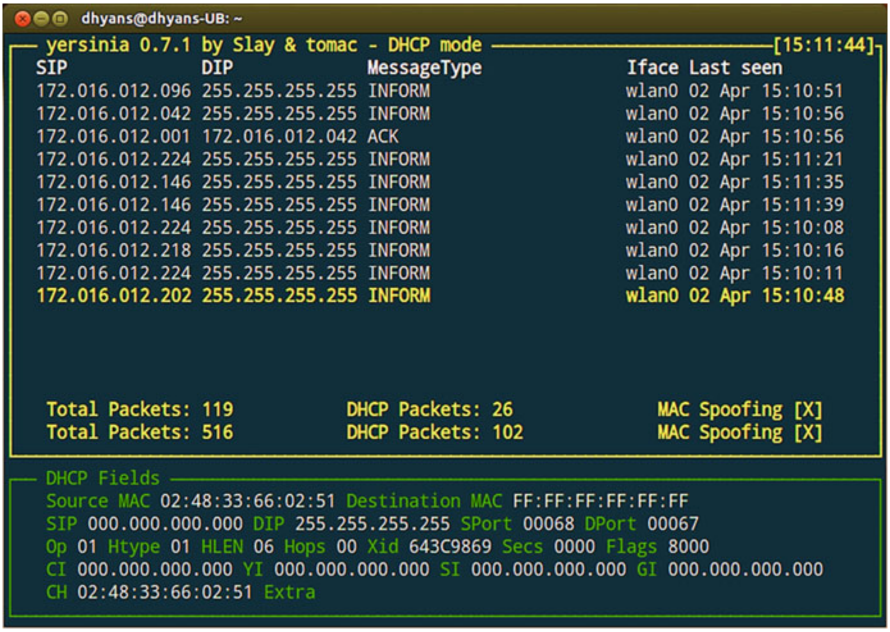
Yersinia

Yersinia [9] is a network security tool for performing layer 2 attacks. It is designed to take advantage of weaknesses of layer 2 protocols for UNIX-like systems. Screenshot is shown in Fig. 4.4.



```
dhyans@dhyans-UB:~$ nikto -h 172.16.2.1
- Nikto v2.1.4
---------------------------------------------------------------------------
+ Target IP:          172.16.2.1
+ Target Hostname:    172.16.2.1
+ Target Port:        80
+ Start Time:         2014-04-03 17:07:30
---------------------------------------------------------------------------
+ Server: squid/3.1.6
```

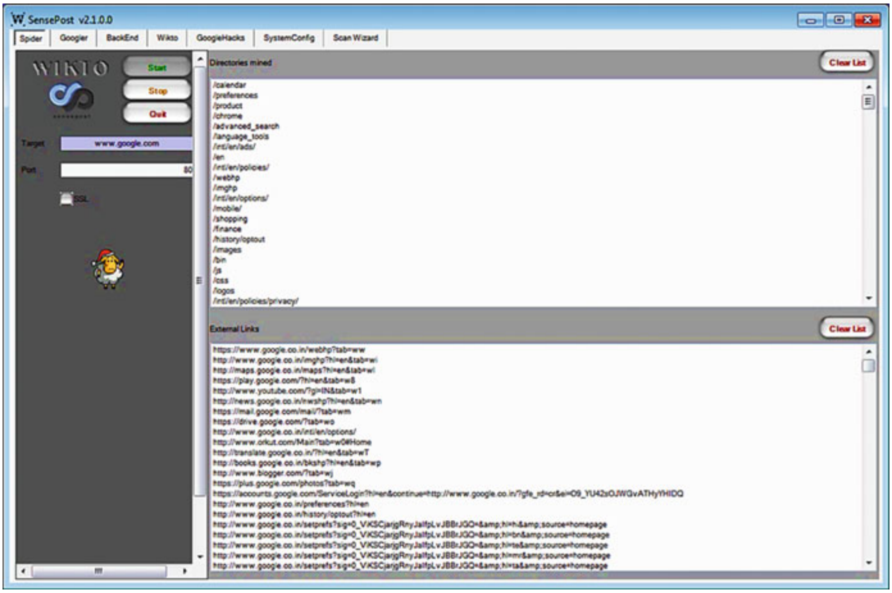
Fig. 4.3 Nikto: scanning for server vulnerabilities



```
┌─ yersinia 0.7.1 by Slay & tomac - DHCP mode ───────────────────[15:11:44]┐
  SIP             DIP             MessageType             Iface Last seen
  172.016.012.096 255.255.255.255 INFORM                  wlan0 02 Apr 15:10:51
  172.016.012.042 255.255.255.255 INFORM                  wlan0 02 Apr 15:10:56
  172.016.012.001 172.016.012.042 ACK                     wlan0 02 Apr 15:10:56
  172.016.012.224 255.255.255.255 INFORM                  wlan0 02 Apr 15:11:21
  172.016.012.146 255.255.255.255 INFORM                  wlan0 02 Apr 15:11:35
  172.016.012.146 255.255.255.255 INFORM                  wlan0 02 Apr 15:11:39
  172.016.012.224 255.255.255.255 INFORM                  wlan0 02 Apr 15:10:08
  172.016.012.218 255.255.255.255 INFORM                  wlan0 02 Apr 15:10:16
  172.016.012.224 255.255.255.255 INFORM                  wlan0 02 Apr 15:10:11
  172.016.012.202 255.255.255.255 INFORM                  wlan0 02 Apr 15:10:48



   Total Packets: 119          DHCP Packets: 26           MAC Spoofing [X]
   Total Packets: 516          DHCP Packets: 102          MAC Spoofing [X]

┌─ DHCP Fields ──────────────────────────────────────────────────────────┐
   Source MAC 02:48:33:66:02:51 Destination MAC FF:FF:FF:FF:FF:FF
   SIP 000.000.000.000 DIP 255.255.255.255 SPort 00068 DPort 00067
   Op 01 Htype 01 HLEN 06 Hops 00 Xid 643C9869 Secs 0000 Flags 8000
   CI 000.000.000.000 YI 000.000.000.000 SI 000.000.000.000 GI 000.000.000.000
   CH 02:48:33:66:02:51 Extra
```

Fig. 4.4 Yersinia: DHCP attack

Yersinia is still under development and supports only a limited number of protocols, such as Dynamic Host Configuration Protocol (DHCP), Dynamic Trunking Protocol (DTP), Spanning Tree Protocol (STP), Cisco Discovery Protocol (CDP), Cisco Inter-Switch Link (ISL), VLAN Trunking Protocol (VTP), Hot Standby Router Protocol (HSRP), IEEE 802.1Q, and IEEE 802.1X.



Target www.google.com
Port 80
□ SSL

Directories mined

```
/calendar
/preferences
/product
/chrome
/advanced_search
/language_tools
/intl/en/ads/
/en
/intl/en/policies/
/webhp
/imghp
/intl/en/options/
/mobile/
/shopping
/finance
/history/optout
/images
/bin
/js
/css
/logos
/intl/en/policies/privacy/
```

External Links

```
https://www.google.co.in/webhp?tab=ww
http://www.google.co.in/imghp?hl=en&tab=wi
http://maps.google.co.in/maps?hl=en&tab=wl
https://play.google.com/?hl=en&tab=w8
http://www.youtube.com/?gl=IN&tab=w1
http://news.google.co.in/nwshp?hl=en&tab=wn
https://mail.google.com/mail/?tab=wm
https://drive.google.com/?tab=wo
http://www.google.co.in/intl/en/options/
http://www.orkut.com/Main?tab=w0#Home
http://translate.google.co.in/?hl=en&tab=wT
http://books.google.co.in/bkshp?hl=en&tab=wp
http://www.blogger.com/?tab=wj
https://plus.google.com/photos?tab=wq
https://accounts.google.com/ServiceLogin?hl=en&continue=http://www.google.co.in/?gfe_rd=cr&ei=O9_YU42sOJWGvATHyYHIDQ
http://www.google.co.in/preferences?hl=en
http://www.google.co.in/history/optout?hl=en
http://www.google.co.in/setprefs?sig=0_VKSCjarjgRnyJalfpLvJBBrJGO=&amp;hl=hi&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_VKSCjarjgRnyJalfpLvJBBrJGO=&amp;hl=bn&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_VKSCjarjgRnyJalfpLvJBBrJGO=&amp;hl=te&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_VKSCjarjgRnyJalfpLvJBBrJGO=&amp;hl=mr&amp;source=homepage
http://www.google.co.in/setprefs?sig=0_VKSCjarjgRnyJalfpLvJBBrJGO=&amp;hl=ta&amp;source=homepage
```

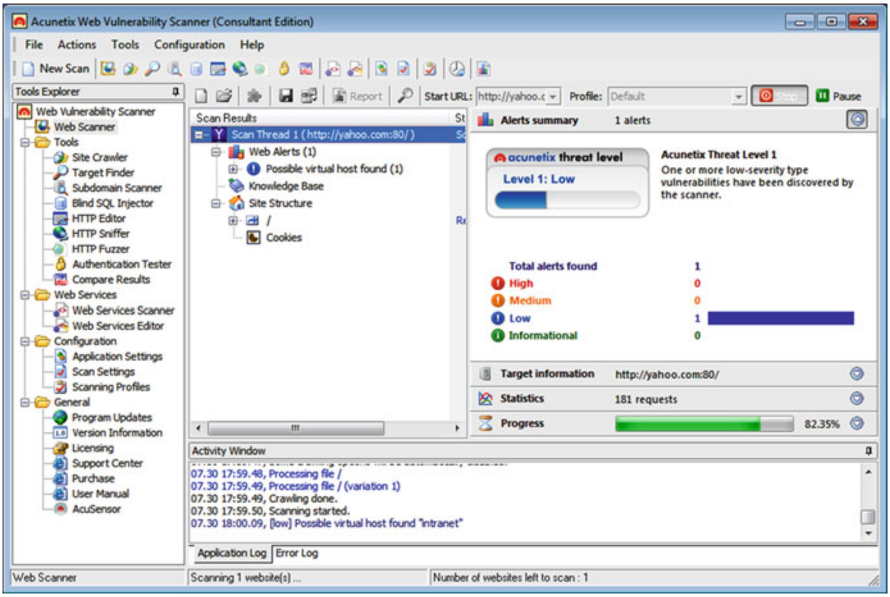
Fig. 4.5 Wikto: a typical web server scan result

Wikto

Wikto [10] is a Web server assessment tool that checks for flaws in Web servers. Wikto scans the Web server to find directories and files stored on that server. It looks for known vulnerabilities and other scripts that can be abused or exploited in the server implementation. Wikto can be seen as Nikto for Windows, only with some extra added features such as a fuzzy logic-based error code checking, back-end mining capabilities, Google-assisted directory mining, real-time HTTP request/response monitoring, etc. Screenshot is shown in Fig. 4.5.

Acunetix Web Vulnerability Scanner

Acunetix Web vulnerability scanner [11] is an application that automatically checks an entire Web site or Web application for security vulnerabilities. Acunetix Web vulnerability scanner is capable of scanning the entire code and scripts for possible vulnerabilities that can be exploited. It also includes some penetration testing tools to automate the whole process. It creates fake attacks and checks the response of the Web site against the attack. After scanning it displays a detailed report of what it has found and how to improve the security. Screenshot is shown in Fig. 4.6.



Start URL: http://yahoo.c
Profile: Default

Alerts summary 1 alerts

Acunetix Threat Level 1
One or more low-severity type vulnerabilities have been discovered by the scanner.

Total alerts found	1
High	0
Medium	0
Low	1
Informational	0

Target information http://yahoo.com:80/
Statistics 181 requests
Progress 82.35%

```
07.30 17:59.48, Processing file /
07.30 17:59.49, Processing file / (variation 1)
07.30 17:59.49, Crawling done.
07.30 17:59.50, Scanning started.
07.30 18:00.09, [low] Possible virtual host found "intranet"
```

Scanning 1 website(s) ...
Number of websites left to scan : 1

Fig. 4.6 Acunetix web vulnerability scanner: a typical web scan

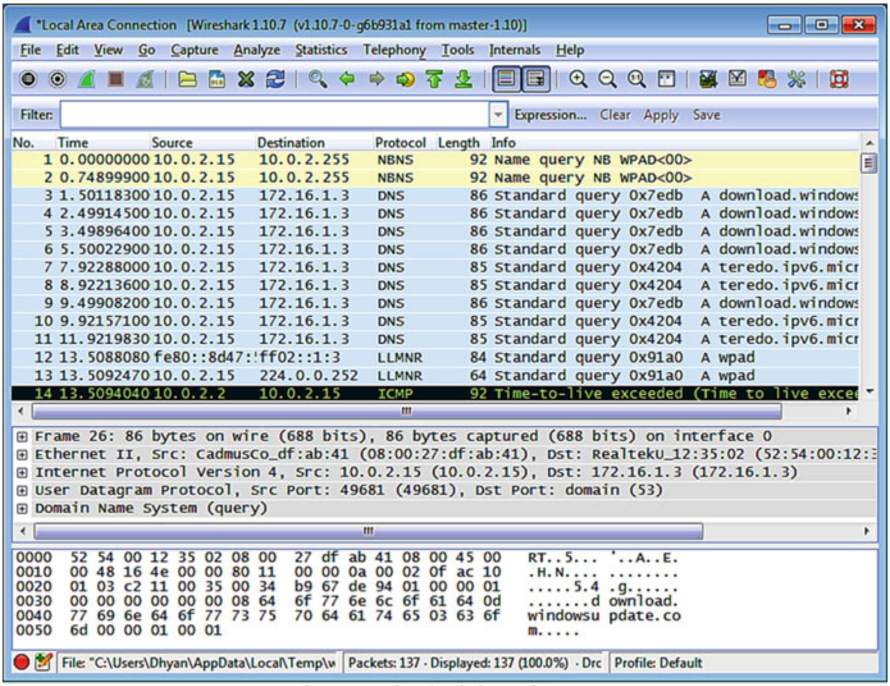
Some features of Acunetix WVS are testing for SQL injection and cross site scripting testing, support for CAPTCHA pages, multithreaded scanning, port scans a Web server and runs security checks against network services running on the server, and advanced penetration testing tools, (e.g., HTTP Editor and the HTTP Fuzzer).

4.4 Network Sniffing and Packet Analyzing Tools

Sniffing and packet analyzing tools include software or hardware that can intercept and capture the data packets passing over a network or a segment of the network. A sniffer captures the data packet and is capable of decoding and showing the various fields of the packet. Packet analyzing tools are used to analyze the captured packet based on the RFC or other standards. Sniffing and analyzing tools help in analyzing network problems, detecting exploitation attempts isolating exploited systems, and monitoring system usage, etc.

Wireshark

Wireshark [12] is an open-source packet and protocol analyzer. Wireshark is cross platform and runs on GNU/Linux, Windows, OS X, BSD, and other UNIX-like


*Local Area Connection [Wireshark 1.10.7 (v1.10.7-0-g6b931a1 from master-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.  Time        Source         Destination   Protocol Length Info
   1 0.00000000  10.0.2.15      10.0.2.255    NBNS      92 Name query NB WPAD<00>
   2 0.74899900  10.0.2.15      10.0.2.255    NBNS      92 Name query NB WPAD<00>
   3 1.50118300  10.0.2.15      172.16.1.3    DNS       86 Standard query 0x7edb  A download.window
   4 2.49914500  10.0.2.15      172.16.1.3    DNS       86 Standard query 0x7edb  A download.window
   5 3.49896400  10.0.2.15      172.16.1.3    DNS       86 Standard query 0x7edb  A download.window
   6 5.50022900  10.0.2.15      172.16.1.3    DNS       86 Standard query 0x7edb  A download.window
   7 7.92288000  10.0.2.15      172.16.1.3    DNS       85 Standard query 0x4204  A teredo.ipv6.micr
   8 8.92213600  10.0.2.15      172.16.1.3    DNS       85 Standard query 0x4204  A teredo.ipv6.micr
   9 9.49908200  10.0.2.15      172.16.1.3    DNS       86 Standard query 0x7edb  A download.window
  10 9.92157100  10.0.2.15      172.16.1.3    DNS       85 Standard query 0x4204  A teredo.ipv6.micr
  11 11.9219830  10.0.2.15      172.16.1.3    DNS       85 Standard query 0x4204  A teredo.ipv6.micr
  12 13.5088080  fe80::8d47::ff02::1:3        LLMNR     84 Standard query 0x91a0  A wpad
  13 13.5092470  10.0.2.15      224.0.0.252   LLMNR     64 Standard query 0x91a0  A wpad
  14 13.5094040  10.0.2.2       10.0.2.15     ICMP      92 Time-to-live exceeded (Time to live exce

⊞ Frame 26: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
⊞ Ethernet II, Src: CadmusCo_df:ab:41 (08:00:27:df:ab:41), Dst: RealtekU_12:35:02 (52:54:00:12:3
⊞ Internet Protocol Version 4, Src: 10.0.2.15 (10.0.2.15), Dst: 172.16.1.3 (172.16.1.3)
⊞ User Datagram Protocol, Src Port: 49681 (49681), Dst Port: domain (53)
⊞ Domain Name System (query)

0000  52 54 00 12 35 02 08 00  27 df ab 41 08 00 45 00   RT..5... '..A..E.
0010  00 48 16 4e 00 00 80 11  00 00 0a 00 02 0f ac 10   .H.N.... ........
0020  01 03 c2 11 00 35 00 34  b9 67 de 94 01 00 00 01   .....5.4 .g......
0030  00 00 00 00 00 00 08 64  6f 77 6e 6c 6f 61 64 0d   .......d ownload.
0040  77 69 6e 64 6f 77 73 75  70 64 61 74 65 03 63 6f   windowsu pdate.co
0050  6d 00 00 01 00 01                                  m.....

File: "C:\Users\Dhyan\AppData\Local\Temp\w | Packets: 137 · Displayed: 137 (100.0%) · Drc | Profile: Default
```

Fig. 4.7 Wireshark in action

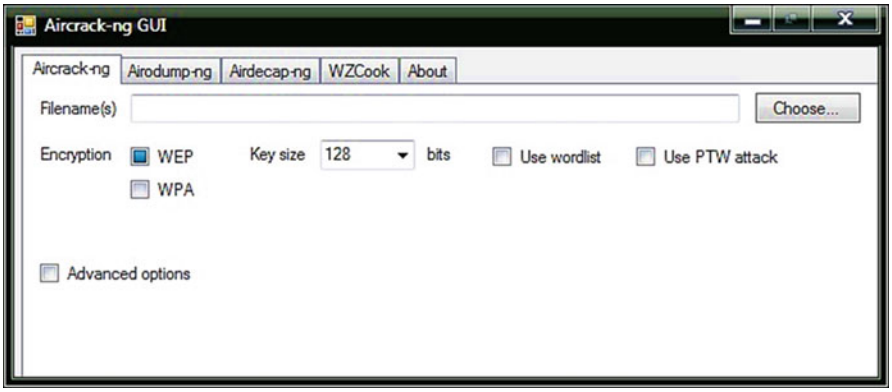
operating systems. Wireshark uses “libpcap” to capture packets from the network hence works only on networks that supports libpcap. Data from an already captured packet can also be read as an input for the analysis. Screenshot is shown in Fig. 4.7.

The main features of Wireshark include:

- Support for Ethernet, IEEE 802.11, PPP, and loopback.
- Live capture and offline analysis.
- Interactive GUI as well as command line version.
- Plug-ins can be created for analyzing new protocols.
- Provides rich VoIP analysis.
- Output can be exported to a number of file formats such as XML, CSV, plain text, and PostScript.

Aircrack-ng

Aircrack-ng [13] is a network software suite for IEEE 802.11 wireless local area networks. Primarily it consists of a detector, packet sniffer, WPA/WPA2-PSK,



Filename(s)

Encryption ■ WEP □ WPA

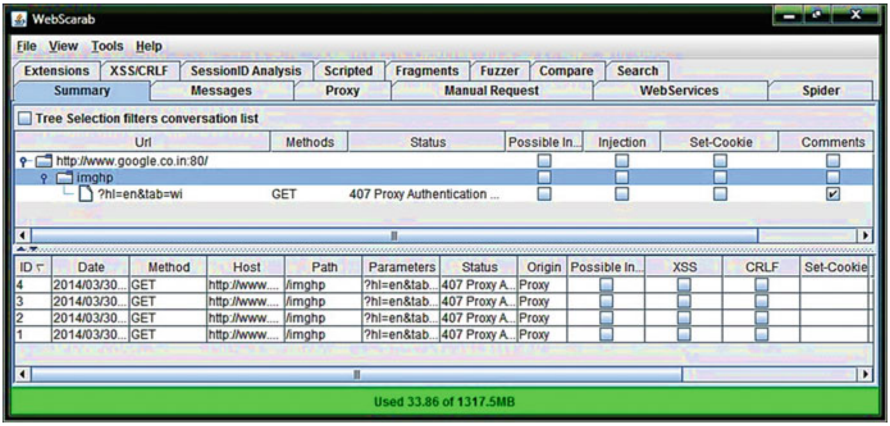
Key size 128 bits

□ Use wordlist

□ Use PTW attack

□ Advanced options

Fig. 4.8 Aircrack-ng GUI for windows



□ Tree Selection filters conversation list

Url	Methods	Status	Possible In...	Injection	Set-Cookie	Comments
http://www.google.co.in:80/			□	□	□	□
imghp			□	□	□	□
?hl=en&tab=wi	GET	407 Proxy Authentication ...	□	□	□	☑

ID	Date	Method	Host	Path	Parameters	Status	Origin	Possible In...	XSS	CRLF	Set-Cookie
4	2014/03/30...	GET	http://www....	/imghp	?hl=en&tab...	407 Proxy A...	Proxy	□	□	□	
3	2014/03/30...	GET	http://www....	/imghp	?hl=en&tab...	407 Proxy A...	Proxy	□	□	□	
2	2014/03/30...	GET	http://www....	/imghp	?hl=en&tab...	407 Proxy A...	Proxy	□	□	□	
1	2014/03/30...	GET	http://www....	/imghp	?hl=en&tab...	407 Proxy A...	Proxy	□	□	□	

Used 33.86 of 1317.5MB

Fig. 4.9 WebScarab: scanning for HTTP/HTTPS traffic

and WEP cracking and analysis tool. Aircrack-ng is available for Windows and Linux and works with any wireless network interface card (NIC) which supports monitoring mode. Screenshot is shown in Fig. 4.8.

WebScarab

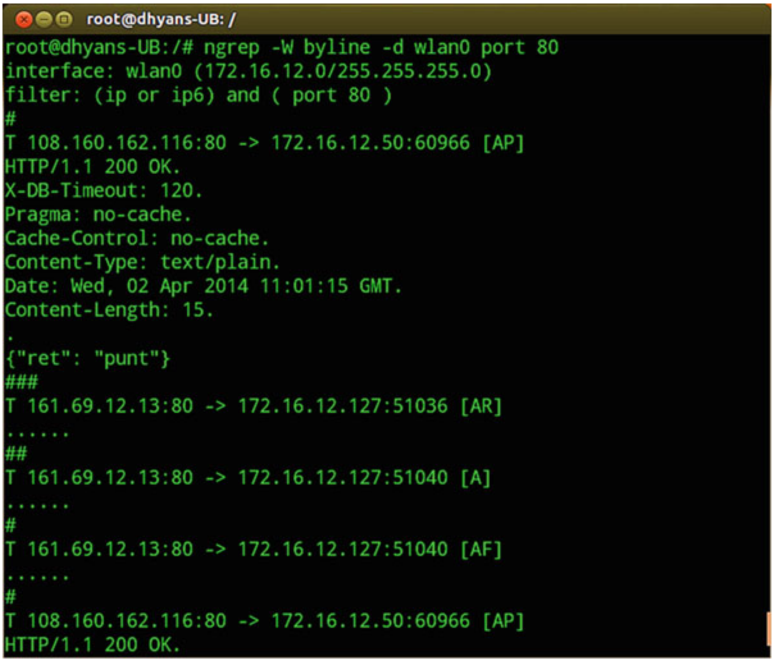
WebScarab [14] is written in Java and analyzes applications that use HTTP and HTTPS protocols for communication. WebScarab operates as an intermediate proxy between the Internet and the application, allowing to review and modify the outgoing requests and incoming responses. Screenshot is shown in Fig. 4.9.

WebScarab has several features in form of plug-ins such as:

- Proxy: To observe the traffic between the application and the Web.
- Manual Intercept: Allows user to modify and requests and responses (HTTP and HTTPS) before they reach to the server.
- Manual Request: Allows manual editing or replaying of previous request or to create entirely new requests.
- Bandwidth Emulator: Allows user to emulate a slower network.

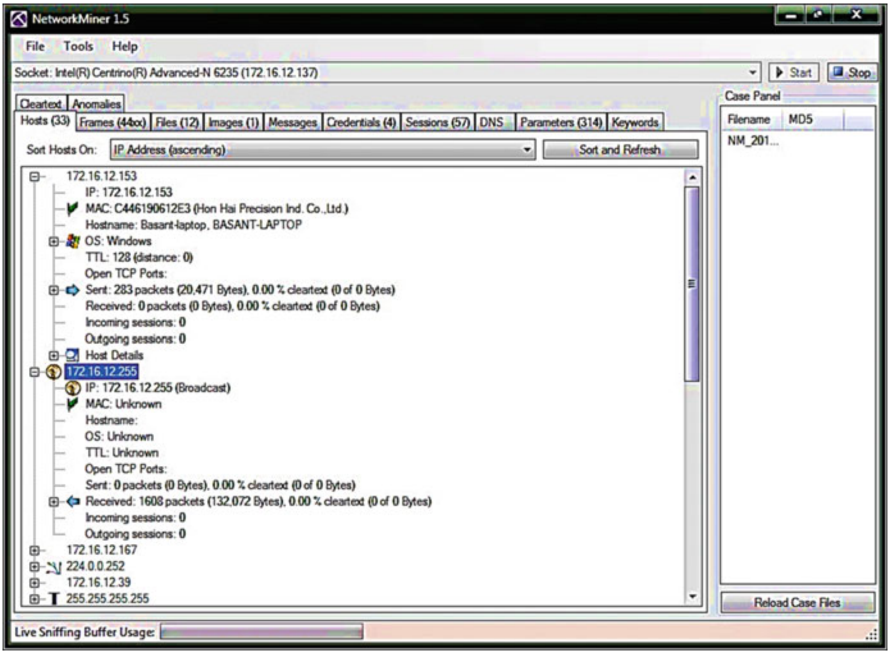
ngrep

"ngrep" [15] is an open-source network packet analyzer for Linux and can be imported to other UNIX-like operating systems. ngrep can look for traffic originating from a specific port and can search for a regular expression in the payload of the packet. ngrep allows user to see all the unencrypted traffic on the network. It supports a wide number of protocols: IPv4 and IPv6, TCP, UDP, ICMPv4 and ICMPv6, IGMP, SLIP, PPP, FDDI, Ethernet, etc. Screenshot is shown in Fig. 4.10.



```
root@dhyans-UB:/# ngrep -W byline -d wlan0 port 80
interface: wlan0 (172.16.12.0/255.255.255.0)
filter: (ip or ip6) and ( port 80 )
#
T 108.160.162.116:80 -> 172.16.12.50:60966 [AP]
HTTP/1.1 200 OK.
X-DB-Timeout: 120.
Pragma: no-cache.
Cache-Control: no-cache.
Content-Type: text/plain.
Date: Wed, 02 Apr 2014 11:01:15 GMT.
Content-Length: 15.
.
{"ret": "punt"}
###
T 161.69.12.13:80 -> 172.16.12.127:51036 [AR]
......
##
T 161.69.12.13:80 -> 172.16.12.127:51040 [A]
......
#
T 161.69.12.13:80 -> 172.16.12.127:51040 [AF]
......
#
T 108.160.162.116:80 -> 172.16.12.50:60966 [AP]
HTTP/1.1 200 OK.
```

Fig. 4.10 ngrep: scanning for traffic generating from port:80



Socket: Intel(R) Centrino(R) Advanced-N 6235 (172.16.12.137)

Sort Hosts On: IP Address (ascending)

```
⊟ 172.16.12.153
   IP: 172.16.12.153
   MAC: C446190612E3 (Hon Hai Precision Ind. Co.,Ltd.)
   Hostname: Basant-laptop, BASANT-LAPTOP
 ⊞ OS: Windows
   TTL: 128 (distance: 0)
   Open TCP Ports:
 ⊞ Sent: 283 packets (20,471 Bytes), 0.00 % cleartext (0 of 0 Bytes)
   Received: 0 packets (0 Bytes), 0.00 % cleartext (0 of 0 Bytes)
   Incoming sessions: 0
   Outgoing sessions: 0
 ⊞ Host Details
⊟ 172.16.12.255
   IP: 172.16.12.255 (Broadcast)
   MAC: Unknown
   Hostname:
   OS: Unknown
   TTL: Unknown
   Open TCP Ports:
   Sent: 0 packets (0 Bytes), 0.00 % cleartext (0 of 0 Bytes)
 ⊞ Received: 1608 packets (132,072 Bytes), 0.00 % cleartext (0 of 0 Bytes)
   Incoming sessions: 0
   Outgoing sessions: 0
⊞ 172.16.12.167
⊞ 224.0.0.252
⊞ 172.16.12.39
⊞ 255.255.255.255
```

Filename	MD5
NM_201...	

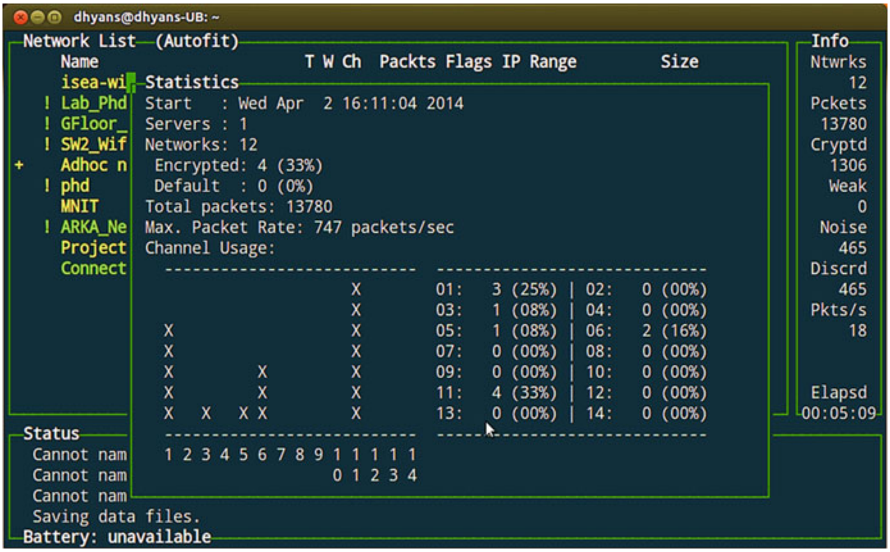
Fig. 4.11 NetworkMiner: scanning for hosts

NetworkMiner

NetworkMiner [16] is a forensic network sniffing tool for Windows and can detect IPs, host names, Operating Systems, and open ports. It can also extract files transmitted over the network. NetworkMiner collects data about the hosts rather than collecting data about the traffic. The main user interface is host centric information grouped per host. Screenshot is shown in Fig. 4.11.

Kismet

Kismet [17] is a wireless network detector and sniffer. Kismet works with any wireless adapter which supports raw monitoring mode (rfmon). Kismet is capable of detecting standard-named (SSID) networks as well as hidden and non-beaconing networks. Kismet passively collects packets without interfering in the network traffic. Screenshot is shown in Fig. 4.12.



```
dhyans@dhyans-UB: ~
Network List—(Autofit)                                              Info
    Name                       T W Ch  Packts Flags IP Range        Size    Ntwrks
    isea-wi Statistics                                                          12
  ! Lab_Phd Start    : Wed Apr  2 16:11:04 2014                             Pckets
  ! GFloor_ Servers : 1                                                      13780
  ! SW2_Wif Networks: 12                                                    Cryptd
+   Adhoc n  Encrypted: 4 (33%)                                               1306
  ! phd      Default  : 0 (0%)                                                Weak
    MNIT    Total packets: 13780                                                 0
  ! ARKA_Ne Max. Packet Rate: 747 packets/sec                                Noise
    Project Channel Usage:                                                     465
    Connect   --------------------------   ----------------------------     Discrd
                                 X         01:   3 (25%) | 02:   0 (00%)       465
                                 X         03:   1 (08%) | 04:   0 (00%)    Pkts/s
              X                  X         05:   1 (08%) | 06:   2 (16%)        18
              X                  X         07:   0 (00%) | 08:   0 (00%)
              X         X        X         09:   0 (00%) | 10:   0 (00%)
              X         X        X         11:   4 (33%) | 12:   0 (00%)    Elapsd
              X   X   X X        X         13:   0 (00%) | 14:   0 (00%)  00:05:09
Status        --------------------------   ----------------------------
  Cannot nam  1 2 3 4 5 6 7 8 9 1 1 1 1 1
  Cannot nam                    0 1 2 3 4
  Cannot nam
  Saving data files.
Battery: unavailable
```

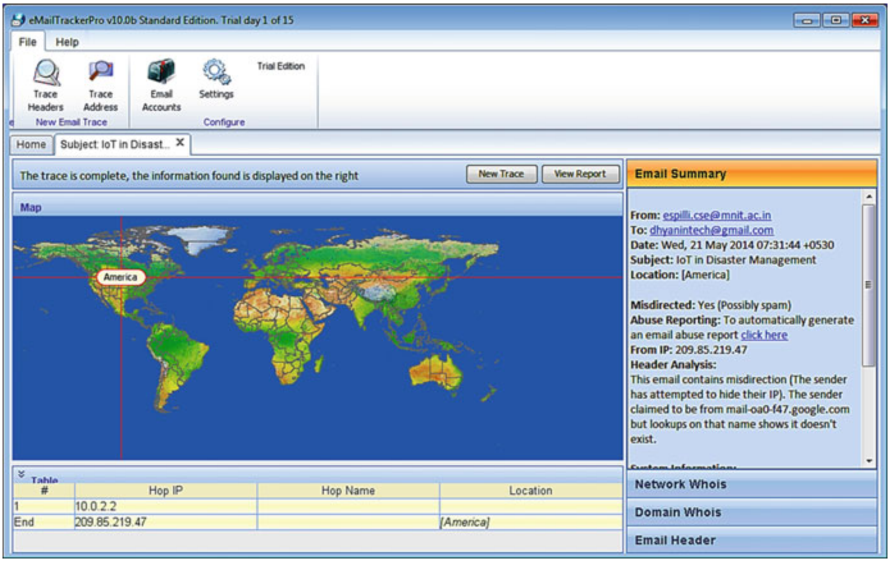
Fig. 4.12 Kismet: network statistics

eMailTrackerPro

eMailTrackerPro [18] offers the ability to trace an e-mail using the e-mail header and the e-mail id itself to its originating location. The advanced version of the eMailTrackerPro also comes with a spam filter, which scans each incoming e-mail for their suspected spam nature. eMailTrackerPro can trace multiple IPs and domains at the same time. Screenshot is shown in Fig. 4.13.

4.5 Network Scanning Tools

Network scanning is a procedure to find the active hosts and the services they offer on a network. Basic scanning procedure involves ping sweep and port scan. Ping sweep returns the information about the active hosts in the networks whereas port scan returns the services a host offers. Sometimes an inverse mapping procedure can also be used to find which IP doesn't correspond to an active host. Network scanning tools provide an automated and efficient way to carry out these tasks.



The trace is complete, the information found is displayed on the right

#	Hop IP	Hop Name	Location
1	10.0.2.2		
End	209.85.219.47		[America]

From: espilli.cse@mnit.ac.in
To: dhyanintech@gmail.com
Date: Wed, 21 May 2014 07:31:44 +0530
Subject: IoT in Disaster Management
Location: [America]

Misdirected: Yes (Possibly spam)
Abuse Reporting: To automatically generate an email abuse report click here
From IP: 209.85.219.47
Header Analysis:
This email contains misdirection (The sender has attempted to hide their IP). The sender claimed to be from mail-oa0-f47.google.com but lookups on that name shows it doesn't exist.

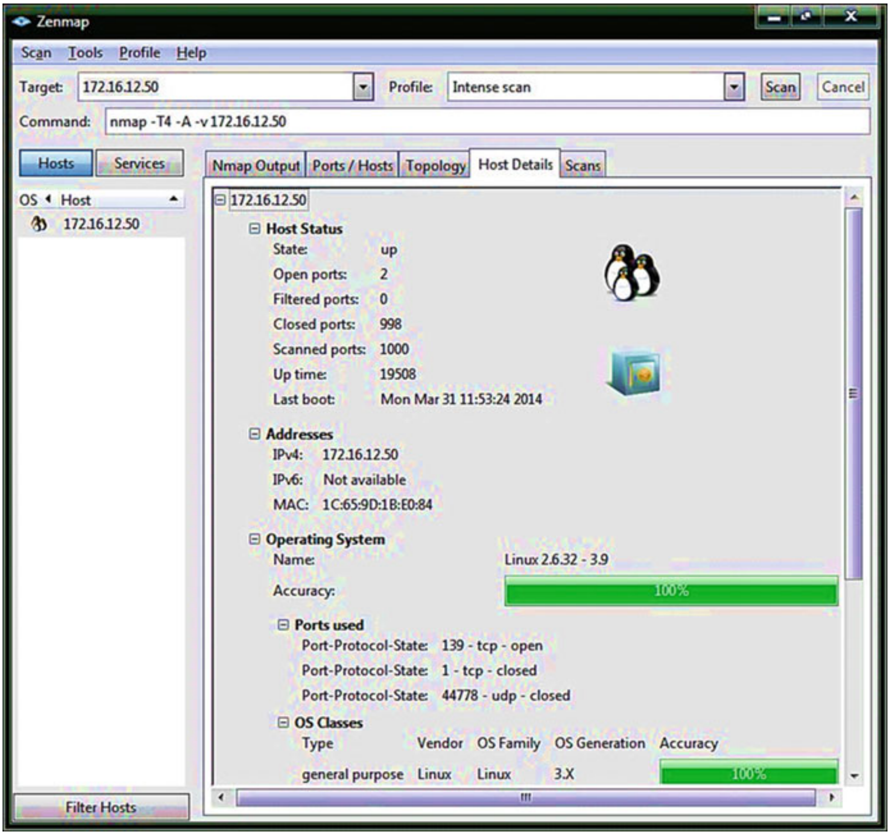
Fig. 4.13 eMailTrackerPro: a typical e-mail scan result

Nmap

Nmap ("Network Mapper") [19] is an open-source, cross platform free network discovery and port scanning tool. It is also useful for mapping an IP to its relative host information. Nmap also offers details about services the hosts' are providing and answers to questions like – what OS is being used? Or what types of packet filter/firewall is being used? etc. Nmap comes with the following options: Zenmap (a GUI result viewer), Ncat (a data transfer and debugging tool), Ndiff (a utility for comparing the scan results), and Nping (packet generation and response analysis tool). Screenshot is shown in Fig. 4.14.

Angry IP Scanner

Angry IP Scanner (or IPScan) [20] is an open-source IP address and port scanner. It can scan IP address in any range by pinging them and then resolving its host name, MAC address, scan ports, etc. It is lightweight and works on cross platform. It does not need to be installed as it can be run from a browser, the only prerequisite being Java Runtime Environment (JRE). The scanning results can be saved in various file formats such as CSV, TXT, XML, and IP-Port list file. IP Scanner uses multithreaded approach to speed up the scanning process. For each IP address, a separate thread is created. The functionality of IP Scanner can be extended using plug-ins. Screenshot is shown in Fig. 4.15.



Target: 172.16.12.50
Profile: Intense scan
Command: nmap -T4 -A -v 172.16.12.50

172.16.12.50

172.16.12.50
Host Status
State: up
Open ports: 2
Filtered ports: 0
Closed ports: 998
Scanned ports: 1000
Up time: 19508
Last boot: Mon Mar 31 11:53:24 2014

Addresses
IPv4: 172.16.12.50
IPv6: Not available
MAC: 1C:65:9D:1B:E0:84

Operating System
Name: Linux 2.6.32 - 3.9
Accuracy: 100%

Ports used
Port-Protocol-State: 139 - tcp - open
Port-Protocol-State: 1 - tcp - closed
Port-Protocol-State: 44778 - udp - closed

OS Classes

Type	Vendor	OS Family	OS Generation	Accuracy
general purpose	Linux	Linux	3.X	100%

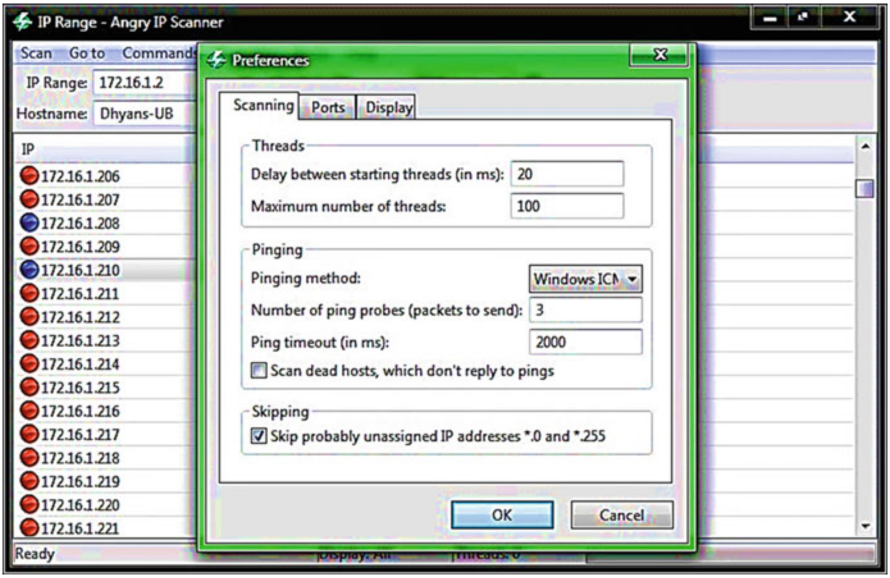
Fig. 4.14 Nmap: scanning for host and port details

Wireless Network Watcher

Wireless network watcher [21] is a small utility by NirSoft, Inc. It scans the wireless network and displays a list of all the nodes and devices that are currently connected to the network. The scan results can also be exported in HTML, CSV, XML, or text format. For every device connected to the network, the following information is displayed: IP address, MAC address, NIC manufacturer, device name, first detection date, and active status. Screenshot is shown in Fig. 4.16.

4.6 Network Monitoring Tools

Computer networks are unreliable, and a host or link may go down any minute affecting the reachability and services provided by the system. Network monitoring tools are generally a collection of simple tools and operating system commands



IP Range: 172.16.1.2
Hostname: Dhyans-UB

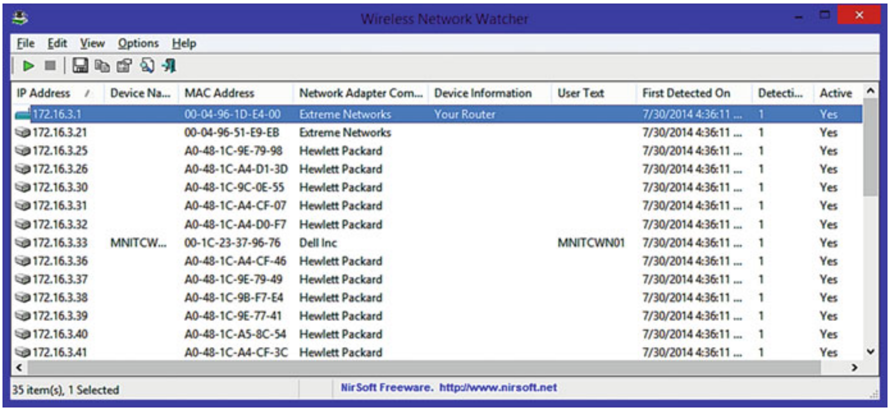
IP
172.16.1.206
172.16.1.207
172.16.1.208
172.16.1.209
172.16.1.210
172.16.1.211
172.16.1.212
172.16.1.213
172.16.1.214
172.16.1.215
172.16.1.216
172.16.1.217
172.16.1.218
172.16.1.219
172.16.1.220
172.16.1.221

Threads
Delay between starting threads (in ms): 20
Maximum number of threads: 100

Pinging
Pinging method: Windows ICM
Number of ping probes (packets to send): 3
Ping timeout (in ms): 2000
☐ Scan dead hosts, which don't reply to pings

Skipping
☑ Skip probably unassigned IP addresses *.0 and *.255

Fig. 4.15 Angry IP scanner: setting the preferences

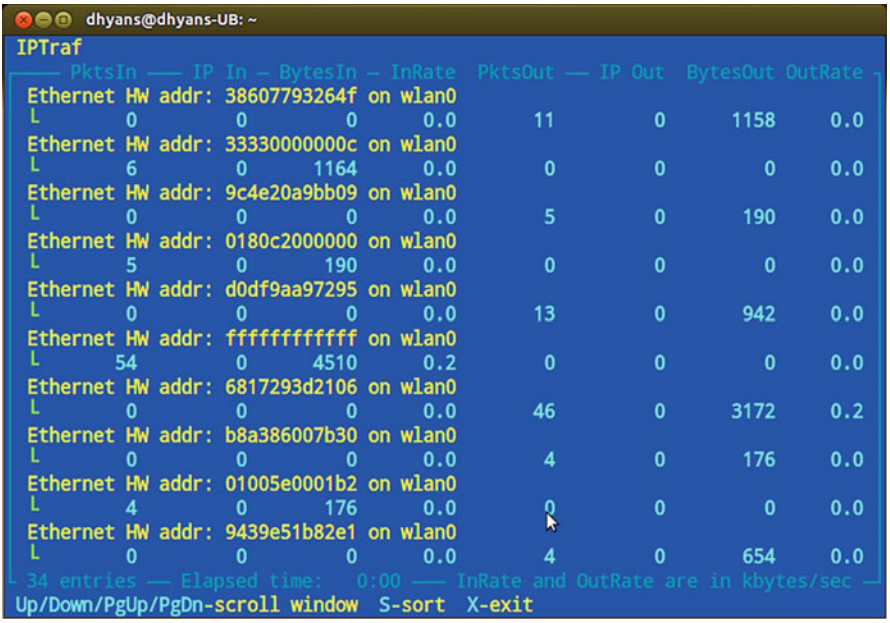


IP Address	Device Na...	MAC Address	Network Adapter Com...	Device Information	User Text	First Detected On	Detecti...	Active
172.16.3.1		00-04-96-1D-E4-00	Extreme Networks	Your Router		7/30/2014 4:36:11 ...	1	Yes
172.16.3.21		00-04-96-51-E9-EB	Extreme Networks			7/30/2014 4:36:11 ...	1	Yes
172.16.3.25		A0-48-1C-9E-79-98	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.26		A0-48-1C-A4-D1-3D	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.30		A0-48-1C-9C-0E-55	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.31		A0-48-1C-A4-CF-07	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.32		A0-48-1C-A4-D0-F7	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.33	MNITCW...	00-1C-23-37-96-76	Dell Inc		MNITCWN01	7/30/2014 4:36:11 ...	1	Yes
172.16.3.36		A0-48-1C-A4-CF-46	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.37		A0-48-1C-9E-79-49	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.38		A0-48-1C-9B-F7-E4	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.39		A0-48-1C-9E-77-41	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.40		A0-48-1C-A5-8C-54	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes
172.16.3.41		A0-48-1C-A4-CF-3C	Hewlett Packard			7/30/2014 4:36:11 ...	1	Yes

35 item(s), 1 Selected
NirSoft Freeware. http://www.nirsoft.net

Fig. 4.16 Wireless network watcher: a typical network scan result

which assist in efficient monitoring of performance, QOS, delay, bandwidth, of a network. Monitoring tools provide a bird's eye view of the entire or a segment of the network.



```
IPTraf
┌─── PktsIn ── IP In ─ BytesIn ─ InRate  PktsOut ── IP Out  BytesOut OutRate ┐
 Ethernet HW addr: 38607793264f on wlan0
  └       0        0        0      0.0       11          0      1158      0.0
 Ethernet HW addr: 33330000000c on wlan0
  └       6        0     1164      0.0        0          0         0      0.0
 Ethernet HW addr: 9c4e20a9bb09 on wlan0
  └       0        0        0      0.0        5          0       190      0.0
 Ethernet HW addr: 0180c2000000 on wlan0
  └       5        0      190      0.0        0          0         0      0.0
 Ethernet HW addr: d0df9aa97295 on wlan0
  └       0        0        0      0.0       13          0       942      0.0
 Ethernet HW addr: ffffffffffff on wlan0
  └      54        0     4510      0.2        0          0         0      0.0
 Ethernet HW addr: 6817293d2106 on wlan0
  └       0        0        0      0.0       46          0      3172      0.2
 Ethernet HW addr: b8a386007b30 on wlan0
  └       0        0        0      0.0        4          0       176      0.0
 Ethernet HW addr: 01005e0001b2 on wlan0
  └       4        0      176      0.0        0          0         0      0.0
 Ethernet HW addr: 9439e51b82e1 on wlan0
  └       0        0        0      0.0        4          0       654      0.0
└ 34 entries ── Elapsed time:   0:00 ─── InRate and OutRate are in kbytes/sec ┘
Up/Down/PgUp/PgDn-scroll window  S-sort  X-exit
```

Fig. 4.17 IPTraf: WLAN monitoring

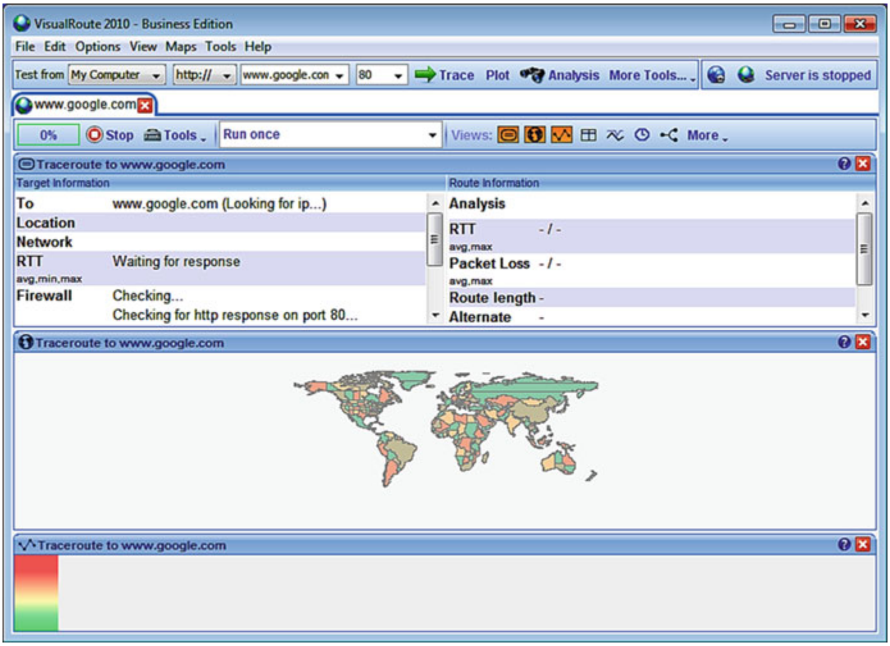
IPTraf

IPTraf [22] is an open-source console-based network statistics utility for UNIX-like systems. It collects a variety of network traffic-related figures such as TCP and UDP packet counts, byte counts, host activities, and interface statistics.

IPTraf supports a number of interfaces such as Ethernet, FDDI, ISDN, SLIP, PPP, etc. IPTraf supports a wide number of packet types: IP, TCP, UDP, IGMP, ICMP, OSPF, IGP, ARP, and RARP. Screenshot is shown in Fig. 4.17.

VisualRoute

VisualRoute [23] is a connectivity diagnostic tool that displays ping and traceroute results in an effective visual form. VisualRoute is primarily used for troubleshooting connectivity issues, performance evaluation of a link on the basis of packet loss, and latency. VisualRoute generates an interactive ping and whois information with time graphs. The results of VisualRoute trace can be exported to a text, HTML reports, or JPG screenshots.


VisualRoute 2010 - Business Edition
File Edit Options View Maps Tools Help
Trace Plot Analysis More Tools...
Server is stopped
Stop Tools
Views: More
Traceroute to www.google.com
Traceroute to www.google.com
Traceroute to www.google.com

Test from My Computer http:// www.google.con 80

www.google.com

0% Run once

Traceroute to www.google.com

Target information

To	www.google.com (Looking for ip...)
Location	
Network	
RTT avg,min,max	Waiting for response
Firewall	Checking... Checking for http response on port 80...

Route information

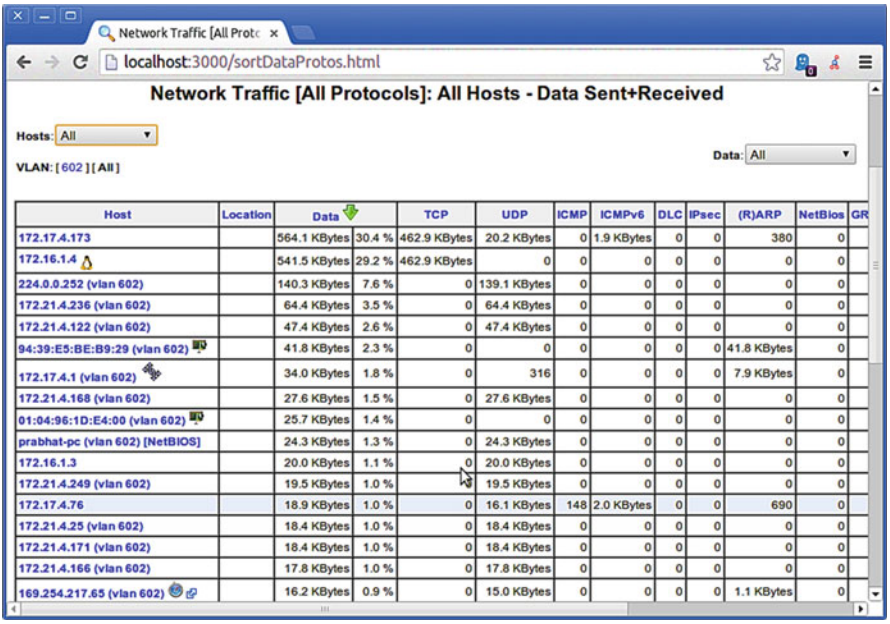
Analysis	
RTT avg,max	- / -
Packet Loss avg,max	- / -
Route length	-
Alternate	-

Fig. 4.18 VisualRoute: looking for a host

The main features of VisualRoute are as follows: IP location reporting, whois lookup, multipath discovery, port testing, port probing, DNS performance testing, traceroute and reverse tracing, etc. Screenshot is shown in Fig. 4.18.

Ntop

Ntop [24] is a network probe that shows network statistics. It has two modes, interactive mode and web mode. In interactive mode, it shows the network status on the terminal. In web mode, it acts as a web server and creates an HTML dump of network status and shows it in the form of Web pages, with graphs and other statistical figures. Ntop can sort traffic according to many protocols, analyze IP traffic, and sort according to the source or destination. It can also geo-locate the hosts. Screenshot is shown in Fig. 4.19.


Network Traffic [All Protc
localhost:3000/sortDataProtos.html

Network Traffic [All Protocols]: All Hosts - Data Sent+Received

Hosts: All

VLAN: [602] [All]

Data: All

Host	Location	Data		TCP	UDP	ICMP	ICMPv6	DLC	IPsec	(R)ARP	NetBios	GR
172.17.4.173		564.1 KBytes	30.4 %	462.9 KBytes	20.2 KBytes	0	1.9 KBytes	0	0	380	0	
172.16.1.4		541.5 KBytes	29.2 %	462.9 KBytes	0	0	0	0	0	0	0	
224.0.0.252 (vlan 602)		140.3 KBytes	7.6 %	0	139.1 KBytes	0	0	0	0	0	0	
172.21.4.236 (vlan 602)		64.4 KBytes	3.5 %	0	64.4 KBytes	0	0	0	0	0	0	
172.21.4.122 (vlan 602)		47.4 KBytes	2.6 %	0	47.4 KBytes	0	0	0	0	0	0	
94:39:E5:BE:B9:29 (vlan 602)		41.8 KBytes	2.3 %	0	0	0	0	0	0	41.8 KBytes	0	
172.17.4.1 (vlan 602)		34.0 KBytes	1.8 %	0	316	0	0	0	0	7.9 KBytes	0	
172.21.4.168 (vlan 602)		27.6 KBytes	1.5 %	0	27.6 KBytes	0	0	0	0	0	0	
01:04:96:1D:E4:00 (vlan 602)		25.7 KBytes	1.4 %	0	0	0	0	0	0	0	0	
prabhat-pc (vlan 602) [NetBIOS]		24.3 KBytes	1.3 %	0	24.3 KBytes	0	0	0	0	0	0	
172.16.1.3		20.0 KBytes	1.1 %	0	20.0 KBytes	0	0	0	0	0	0	
172.21.4.249 (vlan 602)		19.5 KBytes	1.0 %	0	19.5 KBytes	0	0	0	0	0	0	
172.17.4.76		18.9 KBytes	1.0 %	0	16.1 KBytes	148	2.0 KBytes	0	0	690	0	
172.21.4.25 (vlan 602)		18.4 KBytes	1.0 %	0	18.4 KBytes	0	0	0	0	0	0	
172.21.4.171 (vlan 602)		18.4 KBytes	1.0 %	0	18.4 KBytes	0	0	0	0	0	0	
172.21.4.166 (vlan 602)		17.8 KBytes	1.0 %	0	17.8 KBytes	0	0	0	0	0	0	
169.254.217.65 (vlan 602)		16.2 KBytes	0.9 %	0	15.0 KBytes	0	0	0	0	1.1 KBytes	0	

Fig. 4.19 Ntop: network traffic statistics

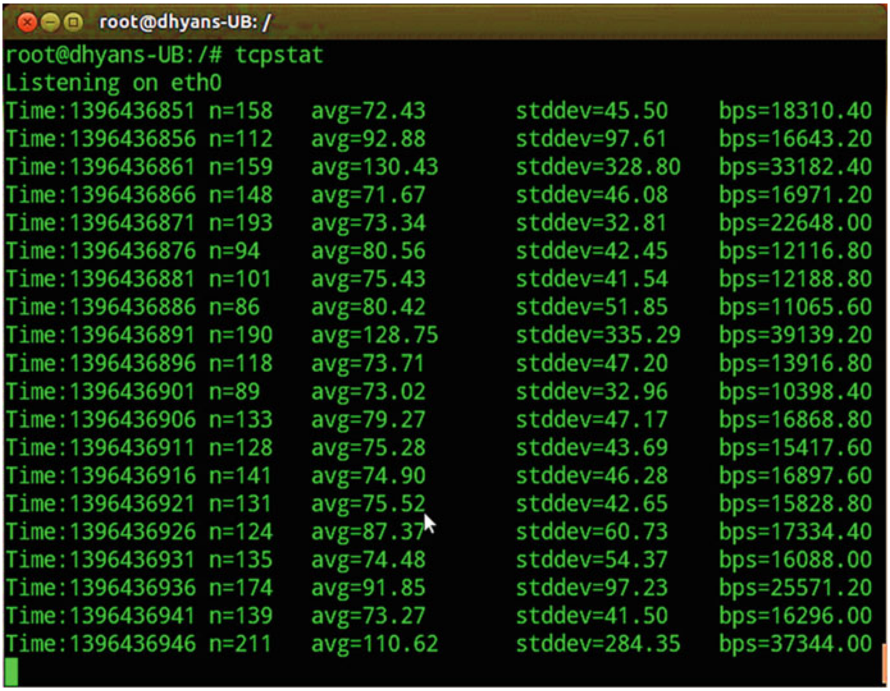
TCPStat

TCPStat [25] reports TCP-related statistics by reading a TCPDump file or by monitoring an interface. It shows statistics like bandwidth, number of packets, average packet size, interface load, standard deviation of packet size, etc. TCPStat is capable of handling a large number of packets per second and has a compact interface. Screenshot is shown in Fig. 4.20.

4.7 Intrusion Detection Systems (IDS)

An intrusion detection system is a software or hardware device which monitors a system or network for potential malicious activities. IDS aim at detecting suspicious traffic and activities originating from inside and outside of the organization.

IDS can either be a network-based (NIDS) or a host-based (HIDS). Some IDS may even try to stop a detected intrusion (intrusion detection and prevention system (IDPS)), but most IDS aim only to detect and report an intrusion.



```
root@dhyans-UB:/# tcpstat
Listening on eth0
Time:1396436851 n=158   avg=72.43       stddev=45.50    bps=18310.40
Time:1396436856 n=112   avg=92.88       stddev=97.61    bps=16643.20
Time:1396436861 n=159   avg=130.43      stddev=328.80   bps=33182.40
Time:1396436866 n=148   avg=71.67       stddev=46.08    bps=16971.20
Time:1396436871 n=193   avg=73.34       stddev=32.81    bps=22648.00
Time:1396436876 n=94    avg=80.56       stddev=42.45    bps=12116.80
Time:1396436881 n=101   avg=75.43       stddev=41.54    bps=12188.80
Time:1396436886 n=86    avg=80.42       stddev=51.85    bps=11065.60
Time:1396436891 n=190   avg=128.75      stddev=335.29   bps=39139.20
Time:1396436896 n=118   avg=73.71       stddev=47.20    bps=13916.80
Time:1396436901 n=89    avg=73.02       stddev=32.96    bps=10398.40
Time:1396436906 n=133   avg=79.27       stddev=47.17    bps=16868.80
Time:1396436911 n=128   avg=75.28       stddev=43.69    bps=15417.60
Time:1396436916 n=141   avg=74.90       stddev=46.28    bps=16897.60
Time:1396436921 n=131   avg=75.52       stddev=42.65    bps=15828.80
Time:1396436926 n=124   avg=87.37       stddev=60.73    bps=17334.40
Time:1396436931 n=135   avg=74.48       stddev=54.37    bps=16088.00
Time:1396436936 n=174   avg=91.85       stddev=97.23    bps=25571.20
Time:1396436941 n=139   avg=73.27       stddev=41.50    bps=16296.00
Time:1396436946 n=211   avg=110.62      stddev=284.35   bps=37344.00
```

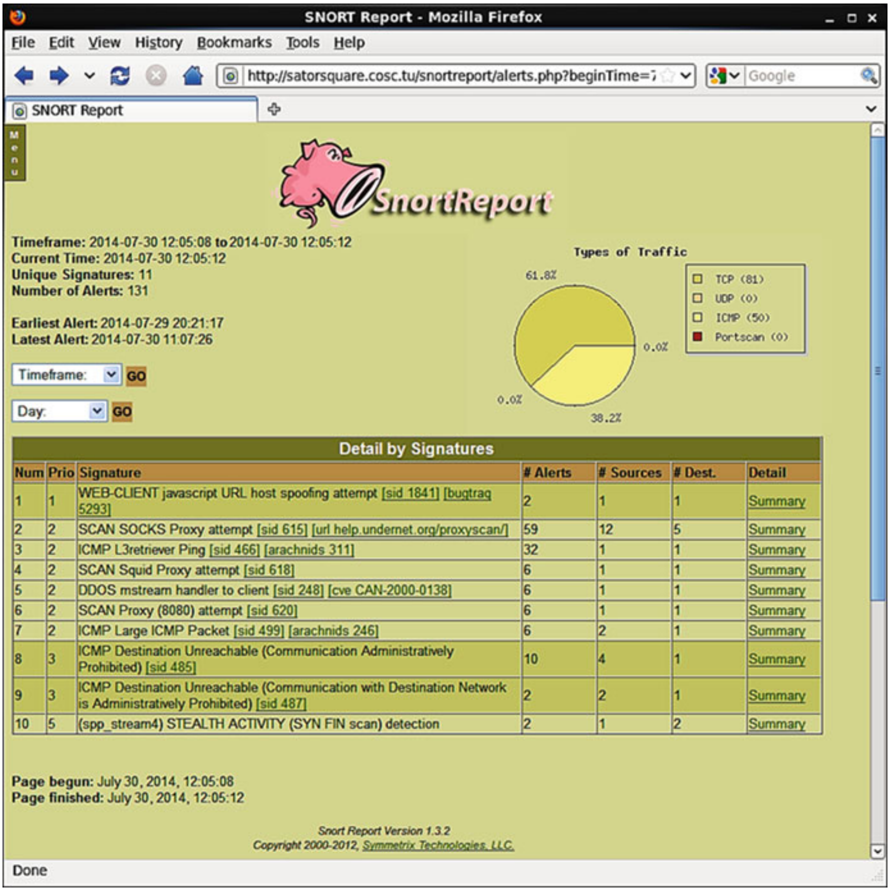
Fig. 4.20 TCPStat: network traffic statistics

Snort

Snort [26] is a network intrusion detection system (NIDS) designed for IP-based networks. Snort analyzes the network traffic and packets to detect worms, vulnerability exploits, port scan, and other suspicious behavior. Snort primarily works in three modes: in sniffer mode, it just reads the network packets and displays on a console; in logger mode, it logs and saves the packets to the disk; and in intrusion detection mode, it analyzes the network traffic against defined rule sets. Snort rules can also be defined by users and checks various attributes of packets whether the traffic should be allowed or blocked. Screenshot is shown in Fig. 4.21.

Bro

Bro [27] is an open-source, passive NIDS for UNIX-like systems. Bro is primarily a network monitor that scans for suspicious link traffic in depth.


SNORT Report - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://satorsquare.cosc.tu/snortreport/alerts.php?beginTime=
Google
SNORT Report
Menu
SnortReport
Types of Traffic
61.8%
0.0%
0.0%
38.2%
□ TCP (81)
□ UDP (0)
□ ICMP (50)
■ Portscan (0)
Timeframe:
GO
Day:
GO
Done

Timeframe: 2014-07-30 12:05:08 **to** 2014-07-30 12:05:12
Current Time: 2014-07-30 12:05:12
Unique Signatures: 11
Number of Alerts: 131

Earliest Alert: 2014-07-29 20:21:17
Latest Alert: 2014-07-30 11:07:26

Detail by Signatures

Num	Prio	Signature	# Alerts	# Sources	# Dest.	Detail
1	1	WEB-CLIENT javascript URL host spoofing attempt [sid 1841] [bugtraq 5293]	2	1	1	Summary
2	2	SCAN SOCKS Proxy attempt [sid 615] [url help.undernet.org/proxyscan/]	59	12	5	Summary
3	2	ICMP L3retriever Ping [sid 466] [arachnids 311]	32	1	1	Summary
4	2	SCAN Squid Proxy attempt [sid 618]	6	1	1	Summary
5	2	DDOS mstream handler to client [sid 248] [cve CAN-2000-0138]	6	1	1	Summary
6	2	SCAN Proxy (8080) attempt [sid 620]	6	1	1	Summary
7	2	ICMP Large ICMP Packet [sid 499] [arachnids 246]	6	2	1	Summary
8	3	ICMP Destination Unreachable (Communication Administratively Prohibited) [sid 485]	10	4	1	Summary
9	3	ICMP Destination Unreachable (Communication with Destination Network is Administratively Prohibited) [sid 487]	2	2	1	Summary
10	5	(spp_stream4) STEALTH ACTIVITY (SYN FIN scan) detection	2	1	2	Summary

Page begun: July 30, 2014, 12:05:08
Page finished: July 30, 2014, 12:05:12

Snort Report Version 1.3.2
Copyright 2000-2012, Symmetrix Technologies, LLC

Fig. 4.21 Snort: a typical snort report

Bro records the network activities in log files in high-level terms. Bro can log all HTTP requests along with their requested URI, headers, MIME types, DNS request-responses, SSL certificates, SMTP sessions, and much more.

Bro can be viewed as a platform for traffic analysis. Bro is fully customizable, and new analysis task can easily be added by the means of scripts. Bro comes with a predefined standard library and supports a wide range of features for detecting intrusions.

4.8 Conclusion

Many tools used in network forensics and network security & monitoring have been introduced and their usage has been discussed. Network forensic tools empower the investigator to monitor networks, gather information about the attack traffic, and assist in analyzing the network crime. A collection of network security and monitoring tools were also introduced, which are very helpful for forensics, though they were not created with evidence collection or analysis in mind. Vulnerability assessment tools, sniffing, scanning, packet and protocol analyzers, monitoring tools and IDS systems have been discussed. Most of the tools discussed are open source and can be installed easily. Though the list of tools is not exhaustive, the essential ones have been introduced, which will suffice to try a first hand at network crime investigation.

4.9 Questions

Objective Questions

1. NetDetector is a ________ build on __________________ architecture.
2. NetIntercept can be placed in the ___________ or at the _____________.
3. _________ supports Ethernet, Gigabit, 10 Gigabit, 802.11a/b/g/n/ac wireless, VoIP, video, MPLS, and VLAN.
4. PyFlag analyzes captured packets in ____________ format.
5. _______ is capable of reconstructing protocol's application data from captured packets.
6. _______ is basically a penetration testing framework.
7. Nessus is a ___________ tool for Windows, Linux, Solaris, FreeBSD, and MAC.
8. Yersinia is a ___________ tool for performing __________ attacks.
9. ____________ is a Web server assessment tool.
10. Wireshark is an open-source ______ and ______ analyzer.

Short-Answer Questions

1. Compare merits and demerits of NetDetector and NetIntercept.
2. What are the advantages of Metasploit over Nessus?
3. Compare Nikto and Yersinia.
4. Explain in brief Acunetix.
5. Discuss merits and demerits of Wireshark.

Long-Answer Questions

1. List advantage and disadvantage of any three network scanning tools.
2. Discuss merits and demerits of any three network monitoring tools.
3. Discuss any two IDS tools in detail.

References

1. Merlette D, Pruthi DP (2003) Network security; NetDetector: identifying real threats and securing your network. [Online]. Available: https://www.niksun.com/, 09 Dec 2013
2. S Enterprises (2003) Netintercept: a network analysis and visibility tool. [Online]. Available: http://www.sandstorm.com, 15 Dec 2013
3. I N C Wildpackets (2003) OmniPeek Network Analyzer. [Online]. Available: https://www.savvius.com/products/overview/omnipeek_family/omnipeek_network_analysis, 12 Dec 2013
4. Cohen M, Collett D (2005) Python forensic log analysis GUI (PyFlag). [Online]. Available: http://www.pyflag.net, 17 Dec 2013
5. Costa G, De Franceschi A (2012) Xplico Internet Traffic Decoder-Network Forensics Analysis Tool. [Online]. Available: http://www.xplico.org/, 21 Dec 2013
6. L L C Metasploit (2007) The metasploit framework. [Online]. Available: https://www.metasploit.com/, 25 Dec 2014
7. Deraison R (2002) The nessus project. [Online]. Available: http://www.nessus.org, 15 Apr 2015
8. Nikto (2001) Web server assessment tool. [Online]. Available: https://cirt.net/code/nikto.shtml, 19 Sept 2015
9. Yersinia (2007) Network tool. [Online]. Available: http://www.yersinia.net/, 17 Jan 2015
10. Wikto (2008) Web server assessment tool. [Online]. Available: http://sectools.org/tool/wikto/, 19 Feb 2015
11. Acunetix (2007) Web vulnerability scanner. [Online]. Available: http://www.acunetix.com/vulnerability-scanner/, 19 Mar 2015
12. Combs G (2007) Wireshark. [Online]. Available: http://www.wireshark.org/, 14 Dec 2014
13. Otreppe T (2013) Aircrack-ng. [Online]. Available: http://www.aircrack-ng.org/, 22 Jan 2014
14. Dawes R (2011) OWASP WebScarab Project. [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project, 30 Mar 2015
15. Ritter J (2006) ngrep_Network grep. [Online]. Available: http://ngrep.sourceforge.net/, 14 Aug 2014
16. NetworkMiner (2008) Network Forensic Analysis Tool (NFAT). [Online]. Available: http://www.netresec.com/?page=NetworkMiner, 23 Apr 2015
17. Kershaw M (2004) Kismet readme. [Online]. Available: http://www.kismetwireless.net/, 08 Nov 2013
18. Inouye D (2002) EmailTrackerPro 1.2 b. [Online]. Available: http://www.emailtrackerpro.com/ , 16 May 2014
19. Lyon G (2009) Nmap: free security scanner for network exploration & security audits. [Online]. Available: https://nmap.org/, 17 Apr 2015
20. A I Scanner (2004) Network Scanner. [Online]. Available: http://angryip.org/, 19 Sept 2013
21. Sofer N (2001) Wireless network watcher. [Online]. Available: http://www.nirsoft.net/utils/wireless_network_watcher.html, 16 Oct 2014
22. Java GP (2001) Iptraf-an ip network monitor. [Online]. Available: http://iptraf.seul.org/, 17 Apr 2015
23. Visualware (2012) Traceroute and network diagnostic tool. [Online]. Available: http://www.visualroute.com/, 01 Mar 2015
24. Deri L, Suin S (1999) Ntop: beyond ping and traceroute. [Online]. Available, 12 July 2013
25. Herman P (2001) The tcpstat tool. [Online]. Available: http://www.frenchfries.net/paul/tcpstat, 29 June 2014
26. Roesch M (1999) Snort: lightweight intrusion detection for networks. [Online]. Available: https://www.snort.org/, 09 July 2013
27. Paxson V (1999) Bro: a system for detecting network intruders in real-time. [Online]. Available: https://www.bro.org/, 14 Mar 2014

Part II
Techniques

Chapter 5
Network Forensic Acquisition

Learning Objectives

- Realize the importance of Network forensic acquisition
- Understand the various formats for collection and preservation of evidence
- Understand the Packet capture file formats – pcap
- Understand the acquisition of evidence at router level – NetFlow and IPFIX

Network forensics deals with the analysis of the trace and log data of network intrusions captured by the existing network security products and provides useful information to characterize intrusion or misbehavior features. The collected data acts as evidence for incident response and investigation of the crime. Network forensics does not block the network crimes but collects enough evidence of the crime. The monitoring and analysis of data from live systems and networks will become essential to law enforcement as caseloads increase and judicial boundaries blur. Network criminals will be punished for their illegal actions thereby providing a deterrent to online crime [1]. The power of various network security and forensic analysis tools [2] available as open source can be integrated so that the investigator can have an edge over the attacker.

The challenge of the network forensics system is to identify useful network events and choose a minimum representative set that would potentially be an evidence in a variety of cybercrimes [3, 4]. The various security and forensic tools collect data about different attributes and protocol features and log them in different formats. The various attributes being misused at the network and transport layer of the protocol can be identified and analyzed. The information collected in various formats can be fused into a file and analyzed for potential evidence information.

We discuss the features and attributes of different protocols of TCP/IP suite, which are being manipulated by the attacker for compromising a system or network. The tools, TCPDump and Wireshark, have been discussed in Chap. 4 and are the preliminary tools to collect IP packets with embedded transport and application protocol information.

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_5

Network security and monitoring tools are not designed to handle forensic investigations, and the way to achieve this is to capture the entire data packets and analyze them in detail. There are two ways to capture the network traffic. Packets can be captured in (i) libpcap (.pcap) files by running a packet sniffer like TCPDump, and (ii) NetFlow data is collected from routers or switches. These captures help the investigators understand what the attack is, who is behind the attack, when it was launched, where the attacker entered the network and how the network defense was breached.

The packet capture format, pcap, and the library libpcap, on which the pcap file format is built, are discussed in the next section. The next-generation format pcapng, is also introduced. The NetFlow and Internet Protocol Flow Information Export format which monitor the packets at network level are also discussed subsequently.

5.1 TCP/IP Protocol Suite

The TCP/IP protocol suite [5] was designed to provide a simple, efficient, open communication infrastructure in an academic and collaborative environment.

The five layers of the protocol suite are shown in Fig. 5.1 taken from [6]. Attackers use the vulnerabilities in the implementation of the TCP/IP protocol stack and exploit them to launch attacks. Important protocols in each layer are discussed briefly in this section.

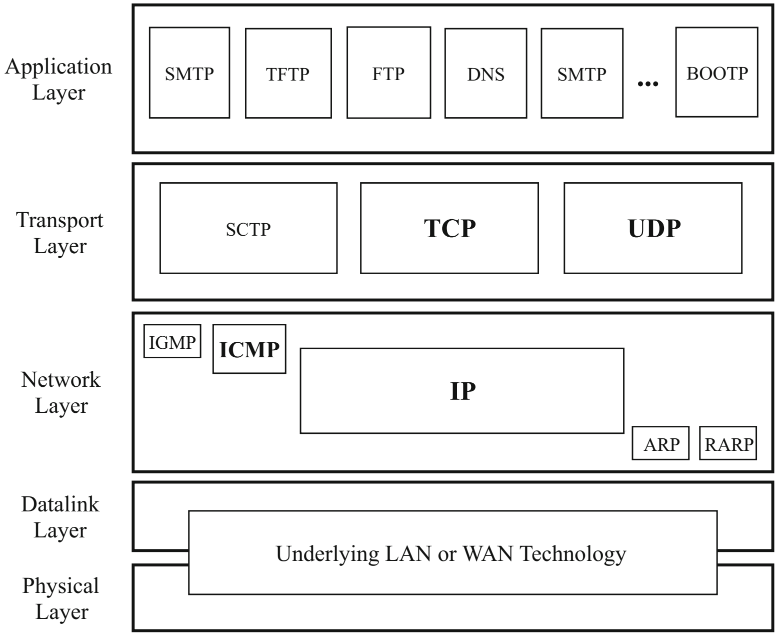


Fig. 5.1 TCP/IP protocol suite

The Internet Protocol (IP) [7] operates at the network layer of the Internet and routes a packet to its destination. The packets go through a series of routers, and at each router, the next hop for the packet is determined. It is possible that two packets from the same source going toward the same destination may take two different paths.

The Internet Control Message Protocol (ICMP) [8] facilitates sending one-way informational message to a host. ICMP is transported in the payload of the IP packet and has several data structures of its own. ICMP is used by a router or a destination host to inform the source host about errors in datagram processing. ICMP allows routers to send error or control messages to other routers or hosts. It also provides communication between the two machines communicating at the network layer. The ICMP protocol is used for two types of operations – reporting non-transient error conditions and probing the network with request and reply messages. ICMP messages are therefore classified into two categories: ICMP error messages and ICMP query messages. Each ICMP message is assigned a number, known as the message type which specifies the type of message. Another number represents a code for the specified ICMP type.

Transmission Control Protocol (TCP) [9] runs on top of IP and provides a connection-oriented service between the source and the destination. TCP provides guaranteed delivery and ensures that the packets are delivered in sequence. It uses various mechanisms, such as sequence numbers, acknowledgments, 3-way handshakes, and timers.

User Datagram Protocol (UDP) [10] is basically an application interface to IP. It provides a mechanism for one application to send a datagram to another. The UDP layer is extremely thin and has low overheads, but it requires that the application takes responsibility for error recovery.

The Hypertext Transfer Protocol (HTTP) [11] is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes, and headers. It is mainly used for accessing data on the World Wide Web (WWW).

Data is transferred between clients and servers using HTTP messages. HTTP messages are read and interpreted by the HTTP server and HTTP client (browser). The format of request and response messages is similar. A request message consists of a request line, header, and a body. Response message has a status line instead of the request line. Request message has many methods for specific actions.

5.2 Packet Capture Format

Network security and monitoring tools are not designed to handle forensic investigations. In order to achieve this, it is required to capture the entire data packets and analyze them in detail. Packets can be captured in libpcap (.pcap) files by running a packet sniffer like TCPDump. These captures help the investigators understand what

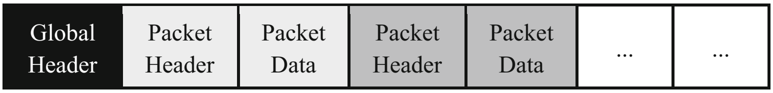


Fig. 5.2 libpcap file format

the attack is, who is behind the attack, when it was launched, where the attacker entered the network and how the network defense was breached.

Libpcap [12] is the very basic file format used to save captured network data. The file extension is .pcap. The file has a global header containing some global information followed by zero or more records for each captured packet as shown in Fig. 5.2. The captured packet in a libpcap file does not contain all the data in the packet as it appeared on the network. It contains at most the first N bytes of each packet. The value of N is called the "snapshot length." N will be a value larger than the largest possible packet to ensure that no packet in the capture is sliced, with a typical value of 65535.

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

The global header is followed by a sequence of packet headers and packet data. The packet header has information fields, *ts_sec*, which gives the date and time when this packet was captured; *ts_usec*, the microseconds offset to *ts_sec* when the packet was captured; *incl_len*, the number of bytes of packet data actually captured and saved in the file; and *orig_len* field which gives the length of the packet as it appeared on the network. The actual packet data will immediately follow the packet header as a data blob of *incl_len* bytes without a specific byte alignment.

The libpcap format is very simple and has gained wide usage. It is however limited in not having time resolution to the level of nanoseconds. It is also not able to display specific connection details, interface information and packet drop count.

5.3 pcapng Dump File Format

pcap Next Generation (NG) [13] is the new format for dumping packet traces to facilitate extensibility, portability and have the ability to merge and append data. A capture file is organized in blocks that are appended one to another to form the file. All the blocks share a common format, which is shown in Fig. 5.3. *Block type* is a unique value that identifies the block. *Block total length* gives the total size of the block in bytes. This field is duplicated at the end for permitting backward file navigation. Content of the block is enclosed in *block body*. Two block types are discussed below which will commonly be found in a.pcapng file. An example file is shown in Fig. 5.4.

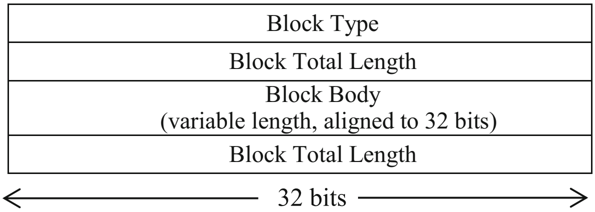


Fig. 5.3 pcapng file format

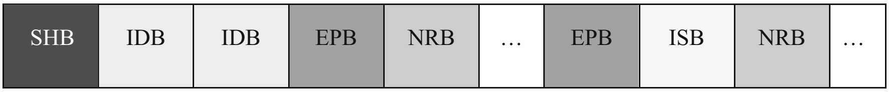


Fig. 5.4 Sample pcapng file structure

The two mandatory blocks which must appear at least once in each file are:

- Section header block (SHB) which defines the most important characteristics of the capture file. It is similar to the global header of libpcap file.
- Interface description block (IDB) which defines the most important characteristics of the interface(s) used for capturing traffic. It contains information on the link layer type of the interface and maximum number of bytes dumped from each packet (snaplen).

The optional blocks which may appear in a file are:

- Enhanced packet block (EPB) which contains a single captured packet or a portion of it. It contains information on interface ID, time stamps, captured length and actual packet length.
- Simple packet block (SPB) which contains a single captured packet, or a portion of it, with only a minimal set of information about it. It does not capture interface ID and time stamps.
- Name resolution block (NRB) which defines the mapping from numeric addresses present in the packet dump and the canonical name counterpart. It avoids the issue of DNS requests every time the capture is opened.
- Interface statistics block (ISB) which defines how to store some statistical data, useful to understand the conditions in which the capture has been made.

5.4 NetFlow Record Format

NetFlow [14, 15] provides network administrators with access to information concerning IP flows within the network. It is deployed for billing, auditing and accounting. It can also be used for intrusion detection, network forensics and

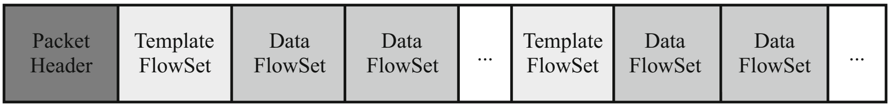


Fig. 5.5 Sample NetFlow export packet

combating DDoS attacks. The basic output of NetFlow is a flow record. The recent evolution of this format is version 9, which is template based, provides extensibility and is future proofed.

NetFlow consists of export packets as shown in Fig. 5.5. The packets are built by a device which has NetFlow services enabled and is addressed to another device which collects and processes the packets. The first part of an export packet is the packet header which provides information about the NetFlow version, number of records, system uptime, UTC seconds, sequence numbering and source ID. *FlowSet* is a collection of records that follow the packet header in an export packet. They are of two types, template and data.

Template FlowSet record defines the format of subsequent data records received in future export packets. FlowSet ID differentiates the template records from data records. Template records have values between 0 and 255 and help in processing NetFlow data without necessarily knowing the data format in advance. Each template record can be distinguished by a unique number called *template ID*. *Length* field gives the total length of the FlowSet. *Field Count* gives the number of fields in this template record. *Field Type* is given by a numeric value that represents the type of field which is vendor specific. Cisco provides values that are consistent across all platforms.

Data FlowSet record provides information about an IP flow that exists on the device that produced an export packet. Each data FlowSet references a previously transmitted template ID. Data record has a FlowSet ID greater than 255. *Record N – Field N* specifies a collection of field values. The type and length have been specified in the template record using the *Field Count*.

5.5 Internet Protocol Flow Information Export (IPFIX) Format

The IPFIX [16, 17] is a standard developed for flexible export of IP flow data from routers or other metering processes, providing network administrators with access to IP flow information. IPFIX is being developed by IETF and is based on Cisco NetFlow V9. It provides record format flexibility through the use of Templates to describe each Data Record. The format is self-describing and facilitates compression, indexing and searching, error recovery, authentication, confidentiality, integrity, anonymization, obfuscation, session auditability, and replayability.

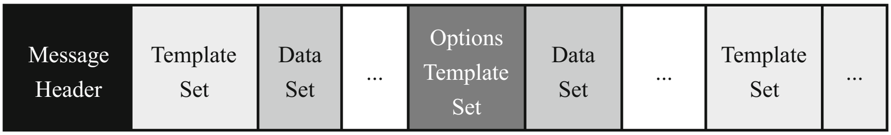


Fig. 5.6 IPFIX message

A *Flow* is defined as a set of IP packets passing an *Observation Point* in the network during a certain time interval. All packets belonging to a particular Flow have a set of common properties. A *Flow Record* contains measured properties of the Flow and characterizes it. A *Metering Process* consists of a set of functions that includes packet header capturing, time stamping, sampling, classifying, and maintaining Flow Records. An *Exporting Process* sends Flow records to one or more *Collecting Processes*.

An *IPFIX Message* originates at the Exporting Process and carries the IPFIX records to a Collecting Process. The format is shown in Fig. 5.6. The terms used are similar to the NetFlow v9 format. The Message consists of a Message Header, followed by one or more Sets. The *Message Header* provides basic information about the message like the version, length of the message, export time, sequence number and *Observation Domain ID. Set* is a generic term for a collection of records that have a similar structure. A *Template Record* defines the structure and interpretation of fields in a *Data Record.* An *Options Template Record* defines the structure and interpretation of fields and how to *scope* the applicability of the Data Record. An *Information Element* encodes independent description of an attribute that may appear in an IPFIX Record. The scope gives the context of the reported Information Elements in the Data Records.

5.6 Conclusion

The first step in investigation of a network crime is collection of evidence of the IP packets. The network forensic acquisition involves gathering data in a secure way ensuring chain of custody so that the analysis results can be replayed in a court of law. Logging evidence is done at two levels – network packet level and at the router level. The file formats of pcap, pcapng, Netflow and IPFIX were discussed. The data logged by these files is passed on to next phase of analysis. Analysis is done using various machine learning or statistical techniques. The information obtained from various file formats can be compressed, fused, and combined to form a new file. The amount of information can be reduced if only few attributes of the data can be logged so that the least amount of data can provide the maximum possible evidence for investigation.

5.7 Questions

Multiple Choice Questions

Select the most suitable answer for the following questions:

1. The following tool can be used for capturing network packets:
   (a) TCPPump
   (b) TCPCopy
   (c) TCPDump
   (d) TCPCap
2. The following is an application layer protocol
   (a) HTTP
   (b) SMTP
   (c) BOOTP
   (d) All of the above
3. Packet capture file format, pcap is based on a library called
   (a) libcap
   (b) libc
   (c) libpcap
   (d) libvmi
4. The advanced file format after pcap is
   (a) pcapnext
   (b) pcapng
   (c) pcapnew
   (d) None of the above
5. Mandatory blocks in pcapng file format are
   (a) SHB and IDB
   (b) SHB and EPB
   (c) IDB and NRB
   (d) SPB and NRB
6. Recent version of NetFlow is
   (a) 7
   (b) 9
   (c) 5
   (d) 10

7. IPFIX stands for

    (a) Internal Protocol Flow Information Export
    (b) Internet Protocol Flow Information Export
    (c) Internet Protocol Flow Intelligent Export
    (d) Internet Protocol Flow Information Expert

8. NetFlow was developed by

    (a) Extreme
    (b) Juniper
    (c) Cisco
    (d) Brocade

9. Template records in a NetFlow record can have values between

    (a) 0 and 255
    (b) 0 and 10
    (c) 0 and 1023
    (d) 0 and 127

10. Optional blocks in pcapng format

    (a) EPB
    (b) SPB
    (c) NRB
    (d) All of the above

Short/Long Answer Questions

1. Compare the similarities and differences in Netflow and IPFIX.
2. List the RFC of the protocols listed in the TCP/IP Suite and write down the basic use of each of the protocol?
3. Discuss the libraries available in Perl or Python for handling pcap or NetFlow file formats.
4. Suggest variations in NetFlow or pcapng file formats to improve the network forensic acquisition.
5. Design a tool to fuse the information from multiple file formats.

References

1. Tang Y, Daniels TE (2005) A simple framework for distributed forensics. In: Proc. 25th IEEE international conference on Distributed Computing Systems Workshops (ICDCS 05), Columbus, OH, USA, pp 163–169
2. Pilli ES, Joshi RC, Niyogi R (2010) Network forensic frameworks: survey and research challenges. Digit Investig 7(1–2):14–27
3. Mukkamala S, Sung AH (2003) Identifying significant features for network forensic analysis using artificial intelligent techniques. Int J Digit Evid 1(4):1–17

4. Shanmugasundaram K, Memon N (2006) Network monitoring for security and forensics. In: Bagchi A, Atluri V (eds) Information systems security, vol 4332. Springer, Berlin/Heidelberg, pp 56–70
5. Leiner B, Rekhter Y (1993) RFC1560: the MultiProtocol internet. Available: http://tools.ietf.org/html/rfc1560, 30 Apr 2011
6. Forouzan BA (2006) TCP/IP protocol Suite, 3rd edn. McGraw Hill Publications, California
7. Postel J (1981) RFC 791: internet protocol. Defense Advanced Research Projects Agency, Available: http://www.ietf.org/rfc/rfc791.txt, 30 Apr 2011
8. Postel J (1981) RFC 792: internet control message protocol. Defense Advanced Research Projects Agency, Available: http://tools.ietf.org/html/rfc0792, 30 Apr 2011
9. Postel J (1981) RFC 793: transmission control protocol. Defense Advanced Research Projects Agency, Available: http://tools.ietf.org/pdf/rfc793.pdf, 30 Apr 2011
10. Postel J (1980) RFC 768: user datagram protocol. Defense Advanced Research Projects Agency, Available: http://tools.ietf.org/pdf/rfc768.pdf, 30 Apr 2011
11. Fielding R, Gettys J, Mogul J, Frystyk H, Masinter L, Leach P, Berners-Lee T (1999) RFC 2616: hypertext transfer protocol – HTTP/1.1. Defense Advanced Research Projects Agency, Available: http://www.ietf.org/rfc/rfc2616.txt, 30 Apr 2011
12. Jacobson V, Leres C, McCanne S (1994) libpcap. LBNL, Berkeley. Available: http://wiki.wireshark.org/Development/Libpcap FileFormat, 30 Apr 2011
13. Degioanni L, Risso F, Varenni G (2004) PCAPNG File Format, IETF, Available: http://www.winpcap.org/ntar/ draft/PCAP-DumpFileFormat.html, 31 Mar 2016
14. Cisco Whitepaper (2007) Cisco IOS NetFlow version 9 flow-record format. Available: http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.pdf, 31 Mar 2016
15. Scheck M (2009) NetFlow for incident detection. In: Proceedings of Forum of Incident Response and Security Teams (FIRST), Kyoto, Japan, pp 1–17
16. Claise B (2008) RFC 5101: specification of the IP Flow Information Export (IPFIX) protocol for the exchange of IP traffic flow information, Available: http://tools.ietf.org/html/rfc5101, 31 Mar 2016
17. Trammell B, Boschi E, Mark L, Zseby T, Wagner A (2009) RFC 5655: specification of the IP Flow Information Export (IPFIX) file format, Available: http://tools.ietf.org/html/rfc5655, 31 Mar 2016

Chapter 6
Network Forensic Analysis

Learning Objectives:

- Background of various machine learning algorithms and network forensics
- Understanding of various phases in a machine learning algorithm
- Understanding of intrusion analysis using classification models

Network forensic analysis is the activity performed by investigators to reconstruct the network activity over a period. The approach is commonly used to investigate individuals suspected of crimes and reconstruct a chain of events during a network-based activity tool. Machine learning is one of the popular approaches used to analyze the network events that adds computer the power to adopt and react according to the situation. These algorithms are used to build models and make predictions based on previous experiences. Michalski et al. [1] precisely defined as a computer program is learning from experience *E* and increasing its performance *P*. Machine learning algorithms are used widely in data mining, pattern classification, medical fields, and intrusion detection for analyzing network traffic. It can be broadly classified into three categories as supervised, unsupervised, and semi-supervised. Supervised machine learning algorithms are those algorithms which are used to learn patterns from labeled input data sets. These algorithms build classifier model from these inputs and then that model is used to classify unknown labels. Algorithms like naive Bayes, decision tree, SVM, and KNN come under this category. In unsupervised algorithms unlabeled data set is provided for building model, and the data is categorized according to similar properties of same class data and different properties for different class data. These algorithms are also called clustering techniques. Examples of unsupervised machine learning algorithms are DBSCAN, one-class SVM, K-means, etc. Semi-supervised algorithm combines the features of both supervised and unsupervised algorithms. They provide high performance for unlabeled data set. Tsai et al. [2] presented review on about machine learning algorithms in network intrusion detection. They had classified machine learning algorithms into four categories, namely, pattern

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_6

classification, single classifiers, hybrid classifiers, and ensemble classifiers. They had reviewed around 55 papers in which different machine learning algorithms are used for constructing IDS. Their review shows that single classifiers are mostly used as baseline classifiers. Hybrid and ensemble approaches are used for increasing accuracy and prediction rate. Some algorithms like genetic algorithm are used mainly for feature selection purpose.

6.1 Misuse Detection

Misuse detection is one of the intrusion detection approaches which is based on making a behavioral profile of known attack patterns and system vulnerabilities. There exist various methods for making intrusion profile of the system. Machine learning is one such method for analyzing the attack patterns of the supplied data set and making generalized profile of the attack traffic.

6.1.1 Naive Bayes

The naive Bayes method was proposed by Thomas Bayes (1702–1761). Bayesian classifiers assign the most likely label to a given example test record defined by its feature vector. Bayesian classifiers are supervised learning approach which follow an assumption that features are independent of a given class, that is, $P(Z|C) = \Pi_{i=1}^{n} P(z_i|C)$; where $Z = (Z1, Z2, ..., Zn)$ represents a feature vector and C is a class or label. The consequence classifier is known as naive Bayes [3]. Let $Z = (z1, ..., zn)$ represents a feature vector, where each feature or attribute belongs to a domain D_i. The set of all feature vectors is denoted $\Omega = D1 \times D2 \times .... \times Dn$. Suppose C be an unknown variable that indicates the class of an example data object, where C can be one of the n values $C \in \{0, 1, ..., n-1\}$. A function $g : \Omega \rightarrow \{0, 1, ..., n-1\}$, where $g(z) = C$, denotes a learning concept. Here, capital letters such as Z_i will denote variables, lowercase letters z_i will denote their values, and bold letters will denote a vector. A deterministic function $h : \Omega \rightarrow \{0, 1, ..., n-1\}$ (a hypothesis) defines a classifier that assigns a class to any given example. Each class is associated with a discriminant function $f_i(z), i = 0, ..., n-1$, and the classifier should be able to select the class with maximum discriminant function on a given example: $h(z) = \arg\max_i \in \{0, 1, ..., n-1\} f_i(z)$. Applying Bayes rule gives $P(C = i|Z = z) = P(Z = z|C = i)\, P(C = i)/P(Z = z)$, where $P(Z = z)$ is similar for all classes, and therefore can be ignored. Thus, Bayes discriminant functions are

$$f_i * (z) = P(Z = z|C = i)\; P(C = i) \tag{6.1}$$

where $P(Z = z|C = i)$ is called the class-conditional probability distribution (CPD). Thus, the Bayes classifier:

$$h * (z) = \arg\max_{i}\ P(Z = z|C = i)\ P(C = i) \tag{6.2}$$

finds the maximum a posteriori probability (MAP) hypothesis for a given example z. When the feature space is high dimensional, then the direct estimation of $P(Z = z|C = i)$ from a given set of training examples is a difficult task. In such cases simple assumption that features are independent for the given class is commonly used. This yields the naive Bayes classifier $NB(z)$ defined by discriminant function:

$$f_iNB(z) = \prod_{j=1}^{n} P(Z_j = z_j|C = i)\ P(C = i) \tag{6.3}$$

Mukherjee et al. [4] defined and classified intrusion detection. The detection of attempts to compromise a computer network resource security is referred to as intrusion detection in the context of information systems. Intrusion detection can be categorized into two general approaches: anomaly detection and misuse detection. Many proposed and implemented techniques for IDS usually generate too many false alerts due to their simple analysis. An attack generally is classified into one of four categories: denial-of-service (dos), probe, user-to-root(u2r), and remote-to-local(r2l).

Amor et al. [5] proposed a new approach based on naive Bayesian networks for intrusion detection. In the probability theory framework, Bayesian networks are useful tools for decision and reason with speculative information. Bayesian networks use directed acyclic graphs to represent casual relations of each node with its parents and conditional probabilities to express the uncertainty of causal relations. A very simple form of Bayes networks is called naive Bayes, based on an assumption that features are independent. Naive Bayesian networks are composed of two levels. At the top level, the root node represents a session class (normal and different kinds of attacks) and, at the bottom level, several leaf nodes, each of them contains a feature of a connection. It is considered to ensure classification that the parent node is a hidden variable which describes the belongingness of each object with a class in the testing data set, and different attributes defining this object are represented by child nodes. The percent of correct classification (PCC) of the instances of the testing set is a parameter to evaluate efficiency and accuracy of classification.

Sharma et al. [6] proposed the layered approach to improve the less significant attack detection rate without disturbing and harming the prediction performance of the majority attacks. The given approach follows naive Bayes classifier on small data set and discretized values for each attack class. In this model, every layer is individually trained and capable of detecting a single type of attack category. In their paper, they defined four layers corresponding to the four attack groups, i.e., probe layer for detecting probe attacks, DoS layer for detecting DoS attacks, U2R for detecting U2R attacks, and R2L layer for detecting R2L attacks.

6.1.2 Decision Tree

Decision Tree (DT) algorithms are used very widely in data mining and machine learning for extracting meaningful information from the input data set. It builds a tree like structure where each branch represents different features and the leaf node represents class labels. According to Safavian et al. [7] DT breaks complex decision, making process into a collection of simpler decision, thus providing a solution which is often easier to interpret. Decision tree the concept of divide and conquer in top-down fashion. The execution flow of decision tree is as:

1. Take the entire set of input.
2. Find the attribute to split that input to maximize purity measure.
3. Divide that input based on splitting attribute. Repeat steps 1–3 for each split.
4. At last pruning is performed to avoid/remove overfitting.

Step 2 as mentioned above is very crucial in decision tree algorithm. Different researchers defined criterion to find splitting attributes. Some famous metrics used for this purpose are information gain, Gini index, and variance reduction. The information gain is the most popular method for finding the best splitting attribute. ID3, C4.5, and C5.0 use this metric for finding more informative attribute for splitting. It is defined as follows:

$$IG(T,a) = H(T) - H(T|a) \tag{6.4}$$

Abbes et al. [8] have used decision tree for developing an intrusion detection system based on protocol analysis. They have used protocol specification file as an input, which can be divided into three parts where the first part is used for defining and initializing the variable, the second part defines detection rules, and the third part defines action rules. The author represented each node of the tree by a tuple (c, R, F, L), where c is the condition, R is the set of candidate detection rule, F is the already used attributes, and L is describing the rules matching at that particular node. Beginning with the root (initial node), the nodes have been recursively decomposed to satisfy different detection rules, until all the detection rules become empty. Decision tree uses this file to learn the behavior of every protocol and develop a classifier model. Further, this model is used for analyzing the protocol behavior.

Stein et al. [9] have applied decision tree algorithm on $KDD'99$ data set for network intrusion detection. In their work, they have used a genetic algorithm for selecting best features to improve detection rate. In experiments, instead of providing an entire input, they have applied genetic algorithm to select different subset of input. By using different subset of features, different decision tree models are built and evaluated through validation data set until the maximum number of generation is considered 100 in their case. After that the best performing classification is used to classify final test data. This whole procedure is used to improve the detection rate of decision tree model. Farid et al. [10] had proposed a new decision tree-based learning algorithm for adaptive intrusion detection. They had used $KDD'99$ data set for building and testing for their model. In their proposed algorithm, initially equal weight W_i is assigned to each instance of input data set which is $1/n$ for n instances in input data set D. After that prior probability $P(C_j)$ is calculated for each class C_j in data set D. Then conditional probability $P(A_{ij}|C_j)$ is calculated for each attribute value in D by using the following as formulae as shown in Eq. (6.5):

$$P(A_{ij}|C_j) = \frac{P(A_{ij})}{\sum\limits_{c_j} W_i} \tag{6.5}$$

where $P(A_{ij})$ is the probability of occurrence of attribute value A_{ij} in class C_j. After that posterior probability P(A_i) of each example in input data set is calculated by the following formulae in Eq. (6.6):

$$P(e_i|C_j) = P(C_j) \prod P(A_{ij}|C_j) \tag{6.6}$$

In the next step, the weight for each instance in input data set is updated with maximum likelihood of posterior probability $P(C_j|e_{(i)})$. Then a decision tree T is constructed and all the above mentioned steps are repeated until all the remaining data belongs to same class or leaf nodes. In experimental setup, they had compared the results of the proposed algorithm with other variants of decision tree algorithms, namely, C4.5 and ID3 for 19 and 41 attributes. They had used detection rate(D.R) and false positive(F.P) as performance evaluation criteria. Their result shows that the proposed algorithms have high detection rate as compared to conventional decision tree methods like ID3 and C4.5.

6.1.3 Nearest Neighbor

Nearest neighbor (NN) is known as lazy classification technique. It simply searches similar or closest instances to classify new observations into their appropriate categories on the basis of the well-known classified observations or training data set. Similarity measure can be calculated using a distance metric-like Euclidean distance [11]. The formula of Euclidean distance between two objects or points, for example, $A1 = (a11, a12, ..., a1\text{n})$ and $A2 = (a21, a22, ..., a2n)$, is as follows

$$\text{distance}(A1, A2) = \sqrt{\sum_{i=1}^{n} (a_{1i} - a_{2i})^2} \tag{6.7}$$

Adetunmbi et al. [11] described that NN is a supervised learning technique which consists of training and testing phases. Data objects are stored in an n-dimensional space with their corresponding labels in the training phase. Unlabeled data are given in the testing phase, and the algorithm calculates a distance metric, and new objects (network traffic) are assigned a label or class of the nearest neighbor with the minimum distance or the most popular label or class in the nearest neighbor (kNN) in the training set. The primary idea of NN was given by Fix and Hodges [12]. Then it has been further improved by Bay [13] and applied in many research-oriented domains, like UCI data sets repository [14]. Many researchers show their interest to use the NN classifier because it classifies objects after computing all possible distance pairs between all the training and the test data set records. Data transformation of continuous attribute values in the range [0, 1] is computed by using a standard technique known as min-max normalization:

$$z' = z - \frac{\min_x}{\max_x - \min_x} \tag{6.8}$$

where $\min_x$ represents the minimum value and $\max_x$ is the maximum value of attribute X. The corresponding value of the attribute in tuple $A1$ with that in tuple $A2$ is compared with each other for categorized attributes. For similar values, the difference between the two will be zero else 1.

Liao et al. [15] proposed a new approach. It classifies program behavior as normal or intrusive based on the k-nearest neighbor (kNN) classifier. Frequencies of system calls represents the program behavior, where each system call is considered as a word and the group of system calls for each program execution as a group of words or a document. The KNN classifier then classified these documents, which is a traditional method of text categorization. The pseudo code for KNN is shown in Algorithm 1.

Algorithm 1 Pseudo-code for KNN

```
Generate normal train $y$ data set (Train_$PB$)
for each process $y$ in test data (Test_$PB$) do
    if $y$ contains an unknown system call then
        $y$ is a abnormal;
    end if
    for each process $x$ in training data set (Train_$PB$) do
        calculate similarity $(y, x)$;
        if similarity $(y, x)$ equals to 1.0 then
            $y$ is normal exit;
        find $k$ biggest scores of similarity $(y, x_j)$;
        calculate similarity_$avg$ for $k$-nearest neighbors;
        end if
        if similarity_$avg$ > threshold then
            $y$ is normal;
        else
            $y$ is abnormal;
        end if
    end for
end for
```

Tsai et al. [16] presented the triangle area-based nearest neighbor (TANN) which is a hybrid learning model to discover attacks more efficiently. In TANN, the very first step is the K-means clustering that repeatedly calculates means of data objects and identifies resulting cluster centers for the attack classes. In the next step, two cluster centers and one data from the given data set are used to calculate the triangle area, and a new feature signature of the data is generated. Based on the newly generated feature signature represented by triangle areas, the *k*-NN classifier now can classify similar attacks. TANN has three phases: the first centers identification using clustering, the second is the new feature signature generation by calculating the triangle area, and the last phase is *k*-NN-based classification on the new data. TANN is an extension to the centroid-based and *k*-nearest neighbor classification approaches. The classification considers all distances between an unlabelled testing data and its nearest centroid and between this unlabeled testing data and other centroids. Thus, in the feature space, any two centroids with an unknown data can result in a triangle area. TANN-based classification is more efficient than the centroid-based and nearest neighbor approaches and gives better performance. Hautamäki et al. [17] proposed two density-based outlier detection methods. According to the first method, if a vector involves in at most T neighborhoods in the kNN graph (a directed proximity graph), then it is defined as an outlier where threshold T is a constraint. In the kNN graph, the vertices of the graph store vectors and edges represent distances between the vectors. A vector can be classified as an outlier by its indegree number of a vector in the graph. According to the second method, all vectors are sorted by their average kNN distances. All the vectors with large average kNN distance are considered as outliers.

6.1.4 Back Propagation Neural Network

A Back Propagation Neural Network (BPNN) is a very widely used neural network algorithm. It has multiple layers and each node has at least one or more interconnected nodes with some "activation function." In back propagation algorithm, the input layer will be at the leftmost side, while the output will be at the rightmost layer, and there may have one or more hidden layers between them. Patterns are presented to the input layer which communicates to the hidden layers, and actual processing takes place through a set of weighted connections. Back propagation works in both forward and backward directions. Initially, forward direction calculation is performed from the input to output layer (through hidden layers), and after that backward calculation is performed in the opposite direction, i.e., from the output to input (by weight updation).

In back propagation the output node is calculated from a number of different regions. It is advisable to use a unary notation to represent the different regions, i.e., for each output only one node can have value 1. Hence, the number of output should be one less than the number of different regions. In this algorithm, every time an input vector of a training sample is presented, the output vector o is compared to the desired value *d*. The comparison is done by calculating the squared difference of the two (1). The value of *Err* tells us how far away we are from the desired value for a particular input. The goal of back propagation is to minimize the sum of *Err* for all the training samples, so that the network behaves in the most "desirable" way (2). We can express *Err* in terms of the input vector (*i*), the weight vectors (*w*), and the threshold function of the perceptions. Using a continuous function (instead of the step function) as the threshold function, we can express the gradient of *Err* with respect to the *w* in terms of *w* and *i*. Given the fact that decreasing the value of *w* in the direction of the gradient leads to the most rapid decrease in *Err*, we update the weight vectors every time a sample is presented using the following formula as given below:

$$Err = (d - o)^2 \tag{6.9}$$

$$\text{Minimize} \sum Err = (d - o)^2 \tag{6.10}$$

$$w_{\text{new}} = w_{\text{old}} - n(\delta Err/\delta w) \tag{6.11}$$

where *n* is the learning rate. Sen et al. [18] have proposed a new back propagation neural network-based intrusion detection system. They have evaluated the performance of this system using different training and testing data sets. They have compared the performance of the proposed system using two evaluation parameters, namely, false-positive rate (FPR) and detection rate (DT). Their result shows that the proposed IDS is performing better than existing systems.

6.1.5 Support Vector Machine

Support vector machine is developed by Corinna et al. [19] in AT & T Bell Labs, Holmdel, which is based on the concept of statistics learning theory [20, 21] and optimal hyperplanes. SVM is designed for a binary classification problem. The basic idea behind SVM is to map input patterns to a high-dimensional feature space *Z*. After that an optimal hyperplane *h* is constructed in that feature space to separate the decision boundaries of different classes. Algorithm 2 describes steps for two-class SVM:

Algorithm 2 Pseudo-code for two-class SVM

```
Initialize the model with training input.
Input is converted to feature space using non-linear transformation function called Kernel Functions
Now next step, is to find best hyper-planes to classify these inputs, a concept of optimal hyperplanes is used in which margin $m$ between the two closest point of different classes is maximized.
Then, input is classified using decision function of hyperplanes like following equations for linear hyperplane:
$w \bullet x + b \geq +1$ for positive class
$w \bullet x + b \leq -1$ for negative class
$w \bullet x + b = 0$ for hyper plane
Where $x$ is input unlabeled instance, $b$ is bias giving distance between origin and hyperplane.
$$\min_{w,b,\xi_i} \frac{\|w\|^2}{2} + C \sum_{i=1}^{n} \xi_i$$
Subjected to
$$y_i(w^t \Phi(x_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, n$$
$$\xi_i \geq 0, \quad i = 1, \ldots, n$$
In the above mentioned way input vectors are classified using support vector machines.
```

As mentioned in Algorithm 2, ξi is used to prevent over-fitting of noisy data in SVM classification function. And $C > 0$ is constantly used for determining trade-off between the number of training inputs within maximum margin. So the final decision function of SVM classifier can be written as multi-class SVM [22]. SVM is designed mainly for binary classification. Due to various complexities, no direct solution is provided for multi-class SVM. Multi-class SVM problem is solved using combination of several binary SVM classifiers. Some popular methods of multi-class SVM are one vs. all SVM and one vs. one SVM and DAGSVM:

1. "One-Against-All" Strategy: In this strategy one SVM is constructed per class, i.e., n SVM is constructed for n number of classes. Along with this all points of a particular class are taken as positive samples and the rest of all the points are taken as negative samples. After that a hyperplane is constructed to distinguish a particular class, i.e., positive sample from the rest of the classes. The key idea behind this strategy is that SVM also predicts a confidence along with class label. Then the class for unknown pattern is predicted based on maximum confidence decision among all SVMs as mentioned in Algorithm 3.

Algorithm 3 Pseudo-code for one-against-all SVM

```
Initialize the model with training input.
for each class label, $k$ do
    I. A new label vector is constructed where $Yi = k$ and all input with class label K are taken
    as positive sample and all other samples are taken as negative samples.
    A list of Classifier $F_k$ is constructed for K number of classes.
end for
Model is built.
Now for an unknown point, all classifiers are applied and point is assigned the class label with
highest confidence.
```

2. "One-Against-One" Strategy: In one vs. one techniques, an SVM is constructed for each pair of classes which means that SVM classifier is trained for data of two particular classes. Suppose we have n classes, then $n * (n - 1)/2$ SVM are designed for that input data set. Classification problem for training data from two classes m and n can be solved using following equations:

$$[\min_{w^{mn},b^{mn},\xi^{mn}} \frac{1}{2}(w^{mn})^T w^{mn} + C\sum_{i=1}^{n} \xi_t^m n \tag{6.12}$$

$$(w^{mn})^T \Phi(x_t) + b^{mn} \geq 1 - \xi_t^m n, \quad if y_t = m \tag{6.13}$$

$$(w^{mn})^T \Phi(x_t) + b^{mn} \geq -1 + \xi_t^{mn}, \quad if y_t = n \tag{6.14}$$

$$\xi_t^{mn} \geq 0 \tag{6.15}$$

So after constructing $n * (n - 1)/2$ classifiers, "majority voting" is mainly used to predict the class for an unknown instance, such as if from one SVM input x is assigned to class m, then one vote is added to that class, and finally the class with the highest number of votes is assigned to that input instance x

3. DAGSVM: It is based on the concept of DAG (directed acyclic graph) which is a treelike structure with no cycles. In DAGSVM, training is done same as in one vs. one classifier, i.e., $n * (n - 1)/2$ classifiers for n classes, but during testing instead of majority voting, a binary DAG is constructed with $n * (n - 1)/2$ non-leaf node and n leaf nodes, where each node resembles an SVM classifier which predicts either m or n class to the input instance. For an unlabeled input x, it starts its traversal from the root, and by passing its path, binary SVM classifier, it reaches to the final leaf node which is the predicted class label for x. DAGSVM has some advantages over other types of multi-class SVMs like some generalization can be established using it. It takes less time as compared to one vs. one class SVM. Mukkamala et al. [23] had given a comparison for intrusion detection systems designed by neural networks and SVM. They had used DARPA data sets for training and testing purpose. They have defined three steps for construction of SVM-based IDS as preprocessing of data sets to extract features from TCP/IP dump PCAP files/traffic, then SVM is trained to learn and classify input data into two classes "normal" or "binary" based on 42 input features extracted

from TCP/IP dump files. Finally testing is done to measure the performance of classifier. Their experimental result shows that both neural networks and SVM had compatible performance with accuracy of around 99 %. But time duration of SVM is much lesser than neural network, i.e., 17.77 s vs 18 min, because neural network requires much more complex calculations for building a classifier model than SVM. Shon et al. [24] had proposed an enhanced SVM for network anomaly detection. Enhanced SVM combines the features of both soft margin (supervised learning) and one-class SVM (unsupervised). So the combined SVM had high detection rate and doesn't require labeled data sets. Equation of soft margin SVM can be written as:

$$\min_{w,b,\xi_i} \frac{\|w\|^2}{2} + C\sum_{i=1}^{l} \xi_i \tag{6.16}$$

subject to:

$$y_i(w^t \Phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \ldots, n \tag{6.17}$$

$$xi_i \geq 0, \quad i = 1, 2, \ldots, n \tag{6.18}$$

And Equation of One class SVM can be written as:

$$\min_{w,b,\xi_i} \frac{\|w\|^2}{2} + \frac{1}{vl}\sum_{i=1}^{l} \xi_i - \rho \tag{6.19}$$

subject to:

$$y_i(w^t \Phi(x_i)) \geq \rho - \xi_i, \quad i = 1, \ldots, n \tag{6.20}$$

$$xi_i \geq 0, \quad i = 1, \ldots, n \tag{6.21}$$

They had used DARPA data set in which the ratio of anomaly traffic as compared to the normal is very less, i.e., the number of outliers is very small. Therefore, one-class SVM not always needs to maximize the distance from the origin for classifying outliers. They had removed bias b from soft margin SVM which considers the distance between the origin and hyperplane. After that by adjusting parameters like C and ρ of one-class SVM and soft margin SVM, they had derived enhanced SVM, as following:

$$\textit{Soft Margin Without Bias} \cong \textit{One Class SVM} \tag{6.22}$$

$$\min_{w,b,\xi_i} \frac{\|w\|^2}{2} + C\sum_{i=1}^{l} \xi_i \cong \min_{w,b,\xi_i} \frac{\|w\|^2}{2} + \frac{1}{vl}\sum_{i=1}^{l} \xi_i - \rho \tag{6.23}$$

$$y_i(w^t \Phi(x_i)) \geq 1 - \xi_i \cong y_i(w^t \Phi(x_i)) \geq \rho - \xi_i \tag{6.24}$$

By assuming ρ as very small like '1'; we can drive enhanced SVM equation as following:

$$\min_{w,b,\xi_i} \frac{\|w\|^2}{2} + C\sum_{i=1}^{l} \xi_i - \rho, \quad \text{where} \quad C \cong 1/vl \tag{6.25}$$

$$y_i(w^t \Phi(x_i)) \geq \rho - \xi_i, \quad 0 \leq \rho \cong \tag{6.26}$$

In experimental setup, they have compared the results of all the three SVM, i.e., one-class SVM, soft margin SVM and enhanced SVM, and the enhanced SVM with the real-world detection system like Bro and Snort. Their test results are showing that enhanced SVM is providing relatively comparable performance. Sung et al. [25] had used SVM and neural networks for selecting features for network intrusion detection. They had used DARPA data set which has 41 features and 5 different class labels, namely, normal, DOS, R2L, U2R, and probing. They had applied rank important features procedure to rank the significance of the input features. Their procedure can be defined as follows: delete a feature from input data set, then use the remaining data set for training and testing purpose, after that analyze the performance of a classifier, and provide ranks to features according to some predefined rules. Repeat the above procedure for all input features. They had defined ten rules to categorize data set features into three categories, namely, important, secondary, and insignificant. These rules are defined on the basis of accuracy, training time, and testing time. After that in experimental setup, they had compared performance of SVM using all the 41 attributes, important features, and union of important features and using important and secondary features. Their results are showing that important features for normal and DOS class are almost same; both U2R and R2L have less important and more secondary features, and the performance of SVM and neural network for all the four conditions of experiments doesn't vary significantly.

6.2 Anomaly Detection

Anomaly detection is another approach for doing the network traffic analysis which is based on making a normal behavioral profile of the traffic. Any deviation from the profile refers to the anomalous behavior of the system. Machine learning is one of the techniques for making a generalized profile of the user's normal behavior.

6.2.1 Self-Organizing Map

It is an unsupervised learning. Everything evolves to the repeated presentation of the pattern, which is basically self-learning that takes place. There are different learning mechanisms but SOM uses competitive learning with some spatial organization.

In the mechanism, several neurons are fully interconnected to the inputs, and out of them, neurons at the output, compete among themselves to determine the winner. This learning mechanism is referred to as "winner-takes-all" mechanism. All the synoptic weights are adjusted in favor of the winning neuron. So, if the same pattern or a pattern very close to the winning pattern is presented again, the chance of that neuron winning the competition improves.

SOM uses competitive learning but there is a spatial organization in the distribution of neurons. It is a lattice of output neuron. In the lattice that can be arranged either as a one-dimensional lattice, two-dimensional lattice, or higher, the neurons can be organized accordingly. One dimensional and two dimensional are used and popular. Higher-dimensional lattices are not so popular because of the complexity that is brought about by them. By organizing the neurons in a structure of a lattice, if those neurons are connected to the input in some manner, then we feed the input pattern, so these input patterns will be actually acting as stimuli to the output neuron. When stimuli is present, then out of the different neurons that are existing in the lattice, one of them will be the winner, and the synaptic connection between the input layer and output layer will be adjusted in such a way (weight updating will take place in such a way) that the Euclidian distance between the input vector and the weight vector is minimized. The minimization of Euclidian distance effectively means the maximization of the WTX output. As we feed various types of input patterns to the system and depending upon input distribution, which is not uniform, one of the neuron emerges as winner. There are two popular models of SOM: (i) Wilshaw-van der Malsburg model in Fig. 6.1 and (ii) Kohonen Network in Fig. 6.2. In Wilshaw-van der Malsburg model, all the pre- and postsynaptic neurons are fully interconnected. This model was used to explain the retina-optic mapping from the retina to visual cortex. The input dimension is the same as the output dimension. Electrical signals of presynaptic neuron are based on geometric proximities. In Kohonen model, the number of inputs can be less than the output. It belongs to vector coding algorithm, and it optimally places a fixed number of vectors (code

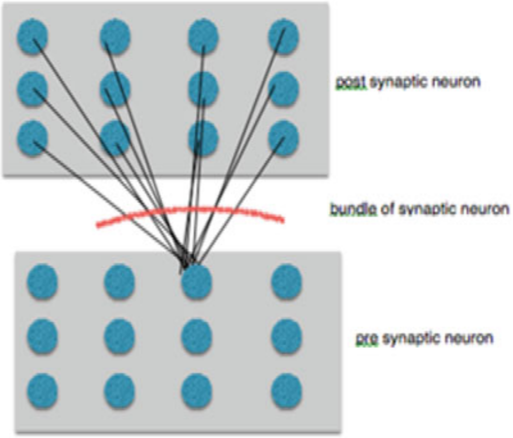


Fig. 6.1 Wilshaw-van der Malsburg model approach

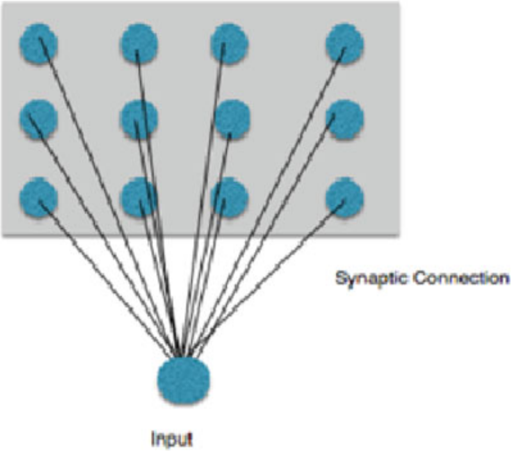


Fig. 6.2 Kohonen model approach

words) into a higher-dimensional input space. Essential processes in the formation of self-organizing maps are as follows:
Competition: Each neuron computes a discriminant function (this function provides basis of competition). The neuron with the largest discriminant function is the winner. **Cooperation:** The winning neuron determines the spatial location of topological neighborhood of the excited neurons. **Synaptic Adaptation:** It enables the excited neurons to increase their individual values of discriminant function in relation to the input pattern.

Vokorokos et al. [26] proposed an intrusion detection system using self-organizing map. The architecture of intrusion detection is based on neural network self-organizing map. The work focuses on modeling user behavioral patterns so they can distinguish between normal and abnormal behavior. System logs were identified and isolated so as to acquire information for the network. The logs provides user activity information and from that, the system derives the following behavioral characteristics, which typifies users on the system: user activity time, user login hosts, user foreign hosts, command set, CPU usage, and memory usage.

Jiang et al. [27] have proposed the application of improved SOM neural network in anomaly detection. The authors have proposed an improved anomaly detection SOM algorithm called FPSOM by introducing learning rate. The experiment shows that the new algorithm performs well and reduces the training time and false-positive rate and effectively improves the detection rate. The idea behind this algorithm is that the weight vector adjustment algorithm should not depend on the number of training cycles; it should depend on the topological structure of the adaptive input data space. FPSOM algorithm makes a performance improvement in terms of detection rate, false alarm rate, and training time over standard SOM. The adjustment of weight vectors in FPSOM avoids it getting trapped in local optimum. In the test of intrusion detection, it shows that the algorithm can effectively reduce the traditional neural network's disadvantages such as longtime training and high false alarm rate. It can also improve the detection rate.

6.2.2 Apriori Algorithm

Association rule is one of the rule-based mining algorithms which became popular in supermarket analysis for finding regularities in shopping behavior of customers. The main goal of the algorithm is to find the frequent patterns and correlation among various items present in the database. The two key factors of Apriori algorithm are support and confidence, which are used to find the most important relationship. For a rule $X \to Y$, support refers to how frequently the items (X and Y) are appearing in the database. Confidence refers to how often item Y appears in those transactions which contain X.

Apriori [28] is an algorithm for finding frequent itemsets using candidate generation [29]. It is characterized as a level-wise complete search algorithm using anti-monotonicity of itemsets. If an itemset is not frequent, any of its superset is never frequent. By convention, Apriori assumes that items within a transaction or itemset are sorted in lexicographic order. Let the set of frequent itemsets of size k be F_k and their candidates be C_k. Apriori first scans the database and searches for frequent itemsets of size 1 by accumulating the count for each item and collecting those that satisfy the minimum support requirement. It then iterates on the following steps and extracts all the frequent itemsets [28] as shown in Algorithm 4.

Algorithm 4 Apriori algorithm

```
$F_i$ = (Frequent itemsets of cardinality 1);
for ($k = 2; F_{k-1} \neq \phi; k{+}{+}$) do
    $C_k$ = apriori-gen ($F_{k-1}$); // New candidates
    for all transection $t \in$ Database do
        $C_t$ = subset ($C_{k+1}, t$);
        Candidates contained in t
        for all candidates $c \in C_t$ do
            (c.count)++;
        end for
        $F_k$ = {C $\in C_k$ | c.count $\geq$ minimum_sup}
    end for
end for
Answer $\cup_k F_k$
```

Apriori is one of the most popular data mining approaches to find frequent itemsets from a transaction data set and derive association rules. Finding frequent itemsets (itemsets with frequency larger than or equal to a user-specified minimum support) is not trivial because of its combinatorial explosion. Once frequent itemsets are obtained, it is straightforward to generate association rules with confidence larger than or equal to a user-specified minimum confidence. Hanguang et al. [30] used Apriori algorithm, which is the classic of association rules in Web-based intrusion detection system and applies the rule base generated by the Apriori algorithm to identify a variety of attacks and improve the overall performance of the detection system. Apriori algorithm in intrusion detection is valued quite a lot by people these days. The improved Apriori algorithm improves execution time greatly, when the data is small. Traditional methods often take a lot of system resources and, thus, a lot of time. Improved algorithm computes support without traversing the database. However, the algorithm complexity increases with larger dataset, taking up considerable memory and processor resources.

6.2.3 K-Means Clustering

K-means clustering is a clustering analysis algorithm that combines data objects based on their attribute values into K disjoint clusters. Objects with similar feature values are clustered into the same cluster. K is a positive integer number, given in advance, specifying the number of clusters. The following are the steps of the K-means clustering algorithm:

1. Set the number of clusters K.
2. Randomly divide all data objects into K cluster and initialize the K cluster centers, compute clusters means, and verify that all cluster centers/centroid are dissimilar from each other.
3. Repeat over all objects and calculate the distances between the centers and objects of all clusters. And then allocate each data object to the cluster with the nearest center.
4. Recompute the centroids of all changed clusters.
5. Reiterate step 3 until the centroids/centers do not modify.

The K-means clustering is used to generate partitions of a data set automatically. It begins with choosing C initial cluster centroids and then repeatedly polishing them as follows:

- Every data instance x_k is allocated to its nearest cluster center.
- The mean of component instances of a cluster is recalculated which becomes the cluster center/centroid v_i.

The clustering process ends when there is no further modification in the assignment of data objects to clusters. Clustering is an iterative process whose purpose is to minimize the objective function (Eq. 6.27):

$$F(X; U, V) = \sum_{i=1}^{C} \sum_{k=1}^{N} d^2(x_k, v_i) \tag{6.27}$$

$d^2(x_k, v_i)$ is the distance measurement.

Münz et al. [31] presented a novel anomaly detection approach based on the K-means clustering algorithm. The raw data contains network traffic flow records exported by routers and network monitors. For predefined time intervals and service-specific port numbers, a transformation of flow records is performed into data sets with a small number of features. The above process is done to identify time intervals showing anomalous traffic behavior. Their approach includes three processing steps: the first step is to transform training data that contains flow records of both normal and malicious traffic into feature data sets, the second step is to apply the K-means clustering and make different partitions of the data sets for normal and anomalous traffic, and finally, by simple distance calculations. The resulting cluster centers are employed for fast anomalies detection in new monitoring data. Clustering may follow the assumption that normal data instances build vast and dense clusters, while anomalies or malicious data instances make very small or distinctive clusters.

Muda et al. [32] presented a hybrid learning approach for anomaly detection. According to them, anomaly detection approaches are capable of predicting attacks with high accuracy and high detection rates. But false alarm rate using anomaly detection method is equally high. To achieve high accuracy and detection rate and lower down the false alarm rate, they proposed a combination of two learning techniques. For the first stage in the proposed hybrid learning approach, they used a K-means clustering as a pre-classification component and grouped similar data instances based on their behaviors. Next, using naïve Bayes classifier, they classified the resulting clusters into attack classes as a final classification task. They found that data that has misclassified during the earlier stage might be correctly classified in the subsequent classification stage.

6.2.4 Genetic Algorithm

Benaicha et al. [33] explained that genetic algorithms belong to computational models stimulated by natural evolution. It is motivated by Darwinian's theory of development and reproductive success to optimize many candidate solutions toward a predefined fitness. John Holland originally invented genetic algorithm in the 1960s. GA follows a progression and natural selection process that employs a chromosome-like data structure and selection; recombination and mutation operators are used to generate the chromosomes. Initially GA randomly generate a population of chromosomes, which illustrate all possible solution of a problem. The goodness of each chromosome is calculated using an evaluation function according to the desired solution; this refers as “fitness function.” Mainly three factors, the fitness function, the representation of individuals, the GA parameters, have a significant impact on the effectiveness of the algorithm and also of the applications. In this paper the goal was to build a parser engine for a GA-based intrusion detection system; this system had two modules, each operated at a different stage. A set of rules was generated from the examined data using GA in the training step. The next step was online intrusion detection, which used the previously generated rules to classify the incoming network connections in real time. Li et al. [34] applied a genetic algorithm to intrusion detection, and according to him, simple rules for network traffic can be generated using genetic algorithms [35]. These network traffic rules work as a separation line between normal network connections and malicious connections. These rules are stored in the rule base in the following form:

if condition then act

In the above statement, the condition field refers to a match comparison between the current network connection and the rules in intrusion detection system, such as source and destination IP addresses and duration of the connection, port numbers, the protocol (TCP/IP, UDP network protocols) used, etc., detecting the probability of the presence of an intrusion. The action field describes an operation performed by the security policies of a system, such as stopping the connection, sending an alert message to the system administrator, and logging that message to system audit files. A rule can be defined as:

if the connection contains following information: source IP addr 124.10.5.28; dest IP addr:130.16.216.55; dest port number: 21; connec-tion time: 15.1 seconds then terminate the connection

The above rule explained as follows: The IP address 124.10.5.28 is detected as one of the blacklisted IP addresses by the IDS. If a network has a connection request for the source IP address 124.10.5.28, destination IP address 130.16.216.55, destination port number 21, and connection time 15.1 s, then terminate that connection. Any connection establishment request for the malicious IP address must be rejected. The main purpose of applying GA is to make such rules that match only the malicious connections. The abnormal connections refer to events with a probability of intrusions. In this experiment, a pre-classified data set of the network traffic is used for GA that distinguishes regular network connections from malicious ones. Different network sniffers (a software which record network traffic without harming system) such as Snort (http://www.snort.com) and Tcpdump (http://www.tcpdump.com) are used to collect network traffic. By using experts' knowledge, data set is manually classified, and this pre-classified data set is used for the fitness calculation during the implementation of GA. It is possible to build a bigger data set having rules for IDS, by initiating GA with only a small set of arbitrarily generated rules. Such rules are "good enough" solutions for GA and capable of filtering new network traffic.

6.2.5 *DBSCAN*

Density-based spatial clustering of application with noise (DBSCAN) [36] is a type of clustering algorithm which is designed basically to handle large spatial databases with minimum domain knowledge. It is based on the concept that density of points inside the cluster is more as compared to the points outside cluster or outliers. This algorithm states that each point in the cluster should have some minimum amount of neighbors in a given radius, i.e., density of cluster should be greater than the minimum threshold value. DBSCAN classifies each point into three categories: (i) core points if the number of points in the neighborhood radius (Eps) of a point p is more than the minimum number of points (MinPts), (2) border point which has fewer points than MinPts in the neighborhood (Eps), and (3) a point which is neither core nor border, called as noise point. DBSCAN take a random point p and find all density reachable points from p. By taking the neighborhood radius Eps and minimum number of points MinPts into consideration, DBSCAN tries to develop a cluster using input points. If the density of the cluster is fewer, then they are considered as outliers or noise point.

6.3 Conclusion

Network forensic analysis involves application of machine learning and data mining to examine and learn attack patterns. Supervised, Semi-supervised and Unsupervised algorithms were discussed. Misuse detection and anomaly detection are two broad classes of attack detection. Popular techniques like Naïve Bayes, Decision tree, Nearest neighbour, SVM and others are discussed. Apriori, SOM, K-Means and other anomaly detection mechanisms are also introduced. The challenge is to find out the right combination of algorithms to analyze a specific type of attack. This phase comprises the heart of network forensics and around three quarters of the time is spent in analysis. Choice of the algorithm and efficiency parameters like specificity, sensitivity and accuracy can be compared.

6.4 Questions

Multiple Choice Questions

Select the most suitable answer for the following questions:

1. What is the full form of DBSCAN?

   (a) Density-based spatial clustering of application with noise
   (b) Database spatial clustering of application with noise
   (c) Density-based spatial clustering approach with noise
   (d) None of the above

2. SVM is initially designed for
   (a) Multi-class problems
   (b) One-class problems
   (c) Two-class problems
   (d) All of the above
3. KNN is a ___________ type of learning.
   (a) Supervised
   (b) Unsupervised
   (c) Semi-supervised
   (d) None of the above
4. Pruning in decision tree is used to avoid __________
   (a) Over-fitting
   (b) Complexity
   (c) Misclassification
   (d) None of these
5. SOM in neural networks stands for
   (a) Self-organizing map
   (b) Service-oriented model
   (c) Self-organizing model
   (d) Statistical-oriented map
6. SVM is based on the concept of ____________
   (a) Lines
   (b) Multidimensional planes
   (c) Hyperplanes
   (d) Curves
7. Which of the following is a clustering technique?
   (a) SOM
   (b) SVM
   (c) Naïve Bayes
   (d) K-means
8. SVM refers to
   (a) Support vector model
   (b) Support vector machines
   (c) Support vector methods
   (d) Support vector modules

9. Who is called the father of original genetic algorithm?
   - (a) John Holland
   - (b) Vapnik
   - (c) Charles Dwain
   - (d) Martin Easter
10. DBSCAN algorithm is proposed for ____________ databases.
   - (a) Multimedia databases
   - (b) Temporal database
   - (c) Geographic data
   - (d) Spatial databases

Short-Answer Questions

Write the brief answers of the following questions:

1. How to decide the best machine learning algorithm for analyzing the network traffic data set.
2. "Neural network takes very longer time." If the statement is true, explain the reason.
3. How to decide the best machine learning algorithm for analyzing the network traffic data set.
4. What are the parameters that affect the accuracy of SVM classifier? Explain with the parameter details.
5. How misuse detection techniques are different than anomaly detection techniques?
6. Explain the role of machine learning techniques in doing network forensic analysis.
7. What are training/validation/test sets? What is "cross-validation?" Name one or two examples of cross-validation methods.
8. What is Bayesian learning?
9. What are support vectors?

Long-Answer Questions

Write in detail the answers of the following questions:

1. What is machine learning? Describe the machine learning phases for analyzing the network data in detail?
2. What is principal component analysis (PCA)? Which eigen value indicates the direction of the largest variance? In what sense is the representation obtained from a projection onto the eigen directions corresponding the largest eigen values optimal for data reconstruction?

3. Calculate training time, R-squared values, confusion matrix, precision, recall, and plot ROC curve by applying SVM, Naïve Bayes, and KNN algorithm on publically available network data set like KDD intrusion data set. You can download KDD'99 data set from the following link: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
4. Compare the performance of KNN, SOM, K-means, and DBSCAN on *KDD′*99 network intrusion data set.

References

1. Michalski RS, Carbonell JG, Mitchell TM (2013) Machine learning: an artificial intelligence approach. Springer Science & Business Media, Berlin
2. Tsai C-F, Hsu Y-F, Lin C-Y, Lin W-Y (2009) Intrusion detection by machine learning: a review. Expert Syst Appl 36:11994–12000
3. Rish I (2001) An empirical study of the naive Bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence, Seattle, vol 3(22). IBM, New York, pp 41–46
4. Mukherjee S, Sharma N (2012) Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technol 4:119–128
5. Amor NB, Salem B, Zied E (2004) Naive Bayes vs decision trees in intrusion detection systems. In: Proceedings of the 2004 ACM symposium on applied computing, Nicosia. ACM, pp 420–424
6. Sharma N, Mukherjee S (2012) Layered approach for intrusion detection using naïve Bayes classifier. In: Proceedings of the international conference on advances in computing, communications and informatics, Chennai. ACM, pp 639–644
7. Safavian SR, Landgrebe D (1990) A survey of decision tree classifier methodology. National Aeronautics and Space Administration, Washington, DC
8. Abbes T, Bouhoula A, Rusinowitch M (2004) Protocol analysis in intrusion detection using decision tree. In: International conference on information technology: coding and computing, 2004. Proceedings. ITCC 2004, Las Vegas, pp 404–408
9. Stein G, Chen B, Wu AS, Hua KA (2005) Decision tree classifier for network intrusion detection with GA-based feature selection. In: Proceedings of the 43rd annual Southeast regional conference, Kennesaw, vol 2, pp 136–141
10. Farid DM, Harbi N, Bahri E, Rahman MZ, Rahman CM (2010) Attacks classification in adaptive intrusion detection using decision tree. World Acad Sci Eng Technol 63:86–90
11. Adetunmbi AO, Falaki SO, Adewale OS, Alese BK (2008) Network intrusion detection based on rough set and k-nearest neighbour. Int J Comput ICT Res 2(1):60–66
12. Fix E, Hodges JL (1951) Discriminatory analysis, nonparametric discrimination: consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas
13. Bay SD (1999) Nearest neighbor classification from multiple feature subsets. Intell Data Anal 3(3):191–209
14. Hettich S, Bay SD (1999) The UCI KDD archive. University of California, Irvine. http://kdd.ics.uci.edu, 31 Mar 2016
15. Liao Y, Vemuri VR (2002) Use of k-nearest neighbor classifier for intrusion detection. Comput Secur 21(5):439–448
16. Tsai FC, Lin CY (2010) A triangle area based nearest neighbors approach to intrusion detection. Pattern Recognit 43(1):222–229
17. Hautamäki V, Ismo K, Pasi F (2004) Outlier detection using k-nearest neighbour graph. In: ICPR (3), Cambridge, pp 430–433

18. Sen N, Sen R, Chattopadhyay M (2014) An effective back propagation neural network architecture for the development of an efficient anomaly based intrusion detection system. In: International conference on computational intelligence and communication networks (CICN), 2014, Bhopal, pp 1052–1056
19. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
20. Vapnik V (2013) The nature of statistical learning theory. Springer Science & Business Media, Berlin/New York
21. Vapnik VN, Vapnik V (1998) Statistical learning theory, vol 1. Wiley, New York
22. Vlasveld R (2013) Introduction to one-class support vector machines. http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/. Accessed 31 Mar 2016
23. Mukkamala S, Janoski G, Sung A (2002) Intrusion detection using neural networks and support vector machines. In: Proceedings of the 2002 international joint conference on neural networks, 2002 (IJCNN'02), Honolulu, pp 1702–1707
24. Shon T, Kim Y, Lee C, Moon J (2005) A machine learning framework for network anomaly detection using SVM and GA. In: Proceedings from the sixth annual IEEE SMC information assurance workshop, 2005 (IAW'05), West Point, pp 176–183
25. Sung AH, Mukkamala S (2003) Identifying important features for intrusion detection using support vector machines and neural networks. In: Symposium on applications and the internet, 2003, proceedings, Orlando, pp 209–216
26. Vokorokos L, Balaz A, Chovanec M (2006) Intrusion detection system using self organizing map. Acta Electrotechnica et Informatica 6:1–6
27. Jiang X, Liu K, Yan J, Chen W (2012) Application of improved SOM neural network in anomaly detection. Phys Procedia 33:1093–1099
28. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules. In: Proceedings of 20th international conference on very large data bases, VLDB, Chile, pp 487–499
29. Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H et al (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14:1–37
30. Hanguang L, Yu N (2012) Intrusion detection technology research based on apriori algorithm. Phys Procedia 24:1615–1620
31. Münz G, Li S, Carle G (2007) Traffic anomaly detection using k-means clustering. In: GI/ITG Workshop MMBnet, Hamburg
32. Muda Z, Yassin W, Sulaiman MN, Udzir NI (2011) Intrusion detection based on K-Means clustering and Naïve Bayes classification. In: 7th international conference on information technology in Asia (CITA 11), Sarawak. IEEE, pp 1–6
33. Benaicha ES, Saoudi L, Guermeche B, Lounis O (2014) Intrusion detection system using genetic algorithm. InScience and information conference (SAI), 2014, London. IEEE, pp 564–568
34. Li W (2004) Using genetic algorithm for network intrusion detection. In: Proceedings of the United States Department of Energy Cyber Security Group, Baltimore, pp 1–8
35. Sinclair C, Pierce L, Matzner S (1991) An application of machine learning to network intrusion detection. In: 15th Annual Computer Security Applications Conference (ACSAC '99), 371–377
36. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, Portland, pp 226–231

Chapter 7
Network Forensic Attribution

Learning Objectives

- Background of attribution in forensics
- Definition of attribution and traceback
- IP traceback mechanisms
- Future directions in traceback and attribution

IP traceback [1–5] is an important strategy to contain the ongoing attacks or to investigate and attribute the attacks in the postmortem stage. The traceback mechanism is shown in Fig. 7.1. IP traceback problem is defined as “identifying the actual source of any packet sent across the Internet.” IP traceback techniques are not capable of preventing and mitigating the attack. They can only identify the source of attack packets. However, this information can be used to conduct postmortem investigation of the attack.

The traceback measures are classified as reactive or proactive. A traceback technique is considered reactive when the process is initiated on the fly in response to an attack. Link testing is a reactive technique, for which input debugging [6] and controlled flooding [7] are examples. These methods make use of the large amount of traffic in a DDoS attack and make attack detection decisions while the attack is in progress.

The techniques fail when the attack traffic subsides and hence are not suitable for postmortem analysis. A mechanism is proactive when the traceback information is concurrently generated or stored, as the packets are routed through the network. Proactive measures include packet logging, packet marking (probabilistic and deterministic), hybrid approaches (logging and marking), and AS-level traceback techniques.

Packet logging at key routers facilitates identification of the true origin of attack traffic throughout the Internet. The major problem is the processing and storage resources required at the routers. Snoeren et al. [8] proposed source path isolation

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_7

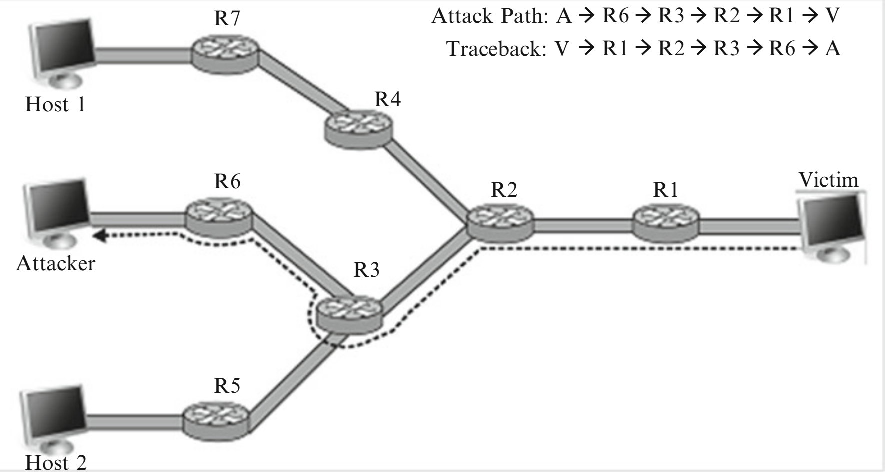

Attack Path: A → R6 → R3 → R2 → R1 → V
Traceback: V → R1 → R2 → R3 → R6 → A
R7
Host 1
R4
R6
R2
R1
Victim
Attacker
R3
R5
Host 2

Fig. 7.1 IP traceback mechanism

engine (SPIE) capable of tracing a single IP packet using packet logging. SPIE system has a centralized traceback manager (STM) to control the data generation agents (DGA) and collection and reduction agents (SCAR). A hash of multiple fields in the IP packet header is computed and logged in the digest tables using space-efficient bloom filters. When a traceback request is made, STM dispatches the information to appropriate SCARs, which query the SPIE-enabled router. STM reconstructs the attack path using the results.

Baba and Matsuda [9] proposed an autonomous management network (AMN), which has a monitoring manager which receives requests from sensors and queries the tracers. Sensors detect the attacks and send the tracing requests. Tracers, implemented in forwarding nodes, maintain log information about incoming packets and their data link-level identifiers. The tracer compares the log data with information about the tracing packet and finds a trace path.

Zhang and Guan [10] proposed a bloom filter-based topology-aware single packet IP traceback system, TOPO, which utilizes router's local topology information for traceback. When a packet travels through the TOPO-enabled router, it records the packet signature and predecessor information. If an attack packet is identified by the victim, the victim's address, packet signature, and packet arrival time are reported to TOPO as a traceback request. All responses from queried TOPO-equipped routers are gathered by TOPO to generate the attack graph. The attack graph is used for further analysis and traceback.

Packet marking involves placing the routers' part or complete address into the IP packet along the attack path. Packets are marked either probabilistically or deterministically. Packets are marked by selecting them randomly with a fixed probability (PPM), or packet may be marked only once by the ingress edge router (DPM).

7.1 Probabilistic Packet Marking

Probabilistic packet marking (PPM) techniques require many packets for convergence of attacker information. Savage et al. [11] proposed PPM where each router receives a stream of packets and probabilistically marks them with partial address information. Packets are marked with a probability p = 0.04 (one in 25 packets). The victim can construct the attack path comprising of all PPM-enabled routers after it has received enough packets. The IP identification field (IP ID) within the IP header is used to store the traceback information. Many variants of PPM have been proposed.

Song and Perrig [12] proposed advanced and authenticated packet marking (AAPM) to further reduce the storage space requirements by encoding the IP address into an 8-bit hash value. It is also assumed that the victim has a complete network map of all upstream routers. When an attack is detected, the marks are extracted, and the attack path is reconstructed by comparing router IP address hashes derived from the network map. Authentication marking scheme based on message authentication codes (MAC) is used to prevent tampering.

Dean et al. [13] proposed algebraic packet marking (APM) that employs algebraic techniques from the field of coding theory to calculate the values of 15-bit marks as points on polynomials. Several schemes like full path encoding, randomized path encoding, and edge encoding are used. Many attack path reconstruction methods are presented. Encoded path information can be stored in the IP fragment ID (16-bit) field of the IP header. Decoding is done by Vandermonde matrix.

Aljifri et al. [14] proposed a simple, novel IP traceback using compressed headers (SNITCH) that is based upon PPM. This technique employs header compression to increase the number of bits available for insertion of traceback information. If an initial frame is sent with a full header, subsequent frames can be sent without the static content (the context) being included in the header.

Yaar et al. [15] proposed fast internet traceback (FIT) that has a packet marking scheme deployed at routers and path reconstruction algorithms used by end hosts. FIT packet markings contain three elements: a fragment of the hash of the marking router's IP address, the number of the hash fragment marked in the packet, and a distance field. Victim uses the hash fragments and distance calculations from the markings in conjunction with its router map.

Deterministic packet marking (DPM) mechanisms are well suited for network forensics as a stream of few packets can sufficiently determine the source of the attacker. They are discussed in detail in the next section. Hybrid mechanisms combine logging and marking of packets. Duwairi and Govindarasu [16] proposed distributed link list traceback (DLLT) based on a "store, mark, and forward" approach. A single marking field is allocated in each packet. Any router that decides to mark the packet stores the current IP address found in the marking field along with the packet ID in a special data structure called marking table maintained at the router, then marks the packet by overwriting the marking field by its own IP address, and then forward the packet as usual. The marking field serves as a pointer to the last router that did the marking for the given packet, and the marking table of that router contains a pointer of the previous marking router.

Jing et al. [17] proposed hierarchical IP traceback system (HITS) with three components for marking, evidence collection, and traceback processing. Each traceback-enabled router has a marking agent (MA) for logging the marking information into its cache or local log database. Traceback Service Provider (TSP) manages the MAs and collects logs from them into a centralized log database. Evidence Collection Agent (ECA) is responsible for collecting marking information as evidence for attacks. The 16-bit ID field and 13-bit offset field are used to encode the marking information, which consists of 8-bit old TTL value and 21-bit hash value of the MAs IP address.

Gong and Sarac [18] developed hybrid single packet IP traceback (HIT) based on marking (append router ID into the marking field) and logging (compute and record packet digest). Traceback-enabled routers audit traffic, and a traceback server having the network topology information constructs attack graph by querying routers. Each router has an ID of 15 bits. The mark is stamped overloading the ID field. The leftmost bit is used as logging flag bit, set to one if router commits logging.

Jing and Lin [19] proposed logging and deterministic packet marking (LDPM) built on a distributed hierarchical IP traceback system. The autonomous system (AS) is considered an independent unit of the Internet. Two kinds of ASs (source and destination) and two kinds of routers (ingress and border) are considered. The goal of LDPM is to trace the special edge connecting ingress and border routers. The 16-bit ID field is used to store the AS ID and 13-bit fragment offset field stores the router ID.

Messaging techniques are also proactive. Bellovin [20] proposed that each router probabilistically selects a packet and generates an ICMP traceback message (iTrace) that is sent to the same destination as the packet. One iTrace message, generated for every 20000 packets, includes the router ED, time stamp, previous and next IP addresses, MAC addresses, and some HMAC authentication data. Intention-driven iTrace [21] is an enhancement to enable the receiver to request for on-demand traceback. This request is received by the upstream routers which set an intention bit in the packet forwarding table.

7.2 Deterministic Packet Marking

Belenky and Ansari [22, 23] first proposed the idea of deterministic packet marking (DPM) where only the ingress edge routers mark the packets, and all other routers are exempt from marking. Each border router marks every packet with its identity before the packet enters the network. DPM uses the 16-bit ID field and the 1-bit reserved field for marking. The IP address of edge routers is split into two segments with 16 bits each. Victim can recover the address once it receives both the segments from the same router. One bit is used as a flag to indicate which portion of the IP address is carried.

Rayanchu and Barua [24] proposed a deterministic edge router marking (DERM) where the entire marking information fits into a single packet. A 16-bit packet ID field is marked with the 16-bit hash of the 32-bit IP address of edge router. The victim has a record table consisting of HashMark and ingress address list so that the IP address for a corresponding hash is identified.

Lin and Lee [25] proposed a robust and scalable DPM scheme where multiple hash functions are used to reduce the probability of address digest collisions. 3 bits are used to distinguish the eight different kinds of marks, and the remaining 14 bits carry partial address information comprising of these marks. The scheme has been designed to send every bit of the IP address at least twice and allows trade-off between false positive rate and false negative rate while considering packet loss due to congestion.

Jin and Yang [26] proposed a DPM-based redundant decomposition (DPM-RD) for IP traceback where the marking field consists of only two sections, information and index. Every ingress edge router decomposes its corresponding IP address into several fragments with neighboring fragments having some redundant bits with each other. The IP ID field is marked with one of the fragments. Redundant decomposition makes the address decomposition more flexible, while decreasing false positives.

Xiang et al. [27] proposed a flexible DPM (FDPM) to find the real source of attacking packets. It adopts a flexible mark length strategy for compatibility to different network environments, and it changes the marking rate according to the load of the participating router by a flexible flow-based marking scheme.

7.3 Autonomous System-Based Traceback

The autonomous system (AS) is a connected group of one or more IP prefixes run by one or more network operators which have a single and clearly defined routing policy [28]. Each AS is identified with a globally unique AS number (ASN) which is used in the exchange of exterior routing information. ASN is a 16-bit integer, assigned and managed by IANA [29].

Paruchuri et al. [30] proposed authenticated autonomous system traceback (AAST) to probabilistically mark packets with AS numbers. Two schemes, AS marking and authenticated AS marking, are presented. The mechanism needs 25 bits for marking, and the TOS (8 bits), ID (16 bits), and unused fragment flag bit are used. Marking is done at AS border routers (ASBR), when a packet is forwarded to a router belonging to another AS. It uses 19 bits (16 bits for ASN and 3 bits for the AS_distance field). Authenticated marking assumes a symmetric key infrastructure in each AS. The 25-bit AS marking field is assigned a cipher text generated as E (ASN || RP, KAS). RP is the 9-bit redundancy predicate set to a hash of source destination address pair.

Gao and Ansari [31] propose autonomous system-based edge marking (ASEM) in which only the ingress edge routers of each AS mark packets with AS number according to certain probability. Packets are not remarked by all other routers. The 32-bit marking information consists of four parts, 16-bit AS_PATH storing the transformed ASPATH information, 1-bit FLAG indicating whether the packet has been marked, 3 bits recording the length of ASPATH, and 12-bit hash function of the IP address. The victim needs to receive only a few packets to reconstruct the attack path. ASPATH attribute provides an ordered list of ASs to be traversed, verifying the path.

Tupakula et al. [32] proposed DoSTRACK that can efficiently deal with the TCP SYN and reflection Distributed Denial of Service (DDoS) attacks. The main aim of the scheme is to prevent the attack traffic at the ingress edge router that is nearest to the source of attack. The egress edge router that is connected to the victim network updates the victim's details (such as 16-bit hash value of the 32-bit IP address) to all other ingress routers within the AS/ISP domain. The egress router validates the traffic that is destined to the victim's network and marks the packet with the unique ID of the ingress router. The ingress edge routers apply ingress filtering on the traffic destined to the victim. Prevention of the attack traffic will be done until the ingress edge routers receive a reset signal from the egress edge router.

Korkmaz et al. [33] consider AS-level deployment of log-based IP traceback and propose AS-level single packet traceback (AS-SPT). It logs packet digests at the border routers of participating ASs and traces a given attack packet toward its origin at the AS level. Each AS-SPT-enabled AS maintains an AS traceback (AST) server that monitors the operation of border nodes logging the packets. When the traceback query arrives from victims, AST queries the border routers and sends the response back with collected information or forward the query recursively to its preceding ASs in case the border routers have not logged the packets.

Castelucio et al. [34] propose an AS-level overlay network that operates on the border routers of an AS and builds an overlay network after exchanging information with BGP. The marking system inserts the routers data into the generalized bloom filter (GBF) of an IP packet. The community attribute in the update messages of BGP is used to group destinations that share the same common characteristics. Marking is done by an exclusive OR operation of 16-bit AS number with 8-bit TTL and 8 MSB of 0's.

7.4 Router and Interface Marking

Interface marking mechanisms consider a router interface as an atomic unit for traceback instead of the router itself. Chen et al. [35] proposed the router interface marking for IP traceback where a RIM-enabled router probabilistically marks each packet with the identifier of one of the hardware interfaces that processed the packet. RIM uses a string composed of locally unique router input IDs as a globally unique identifier of a path. RIM uses 5 bits for distance, 6 bits for XOR, and 6 bits for IID. A router probabilistically marks a packet by resetting the distance field to zero and copying the IID of the packet's incoming interface to both the IID and the XOR fields.

In [36] an improvement of the above technique for attack diagnosis (AD) which is a novel attack mitigation scheme, adopting a divide-and-conquer strategy, activated by the victim after the attack, is detected. The victim instructs the upstream routers to mark the packets deterministically and can traceback one attack source. AD combines the concepts of pushback and packet marking, attack detection is performed near the victim host, and packet filtering is executed close to the attack sources. By instructing its upstream routers to mark packets deterministically, the victim can trace back one attack source and command an AD-enabled router close to the source to filter the attack packets.

Yi et al. [37] proposed DPM with link signature, which marks every packet passing through a router with link signature, which is the digest of the address information of the two adjacent nodes or a random 16-bit value. Each router will participate in marking deterministically and the mark will change. The entire path information is available in each packet and single packet IP traceback is possible.

Peng et al. [38] proposed an enhanced and authenticated DPM where path numbering is used for traceback. There are two types of routers, DPM-enabled and PNM-enabled routers. DPM-enabled routers are deployed at the edge of a subnet to mark each packet traversing them by the incoming interface. PNM-enabled routers are closest to the source of the packet and mark each packet with the path identifiers representing the path linking them to the DPM-enabled routers. The victim can not only detect and filter attacks but can also obtain accurate information by the authenticated marks.

The relation between various traceback mechanisms can be seen in Fig. 7.2.

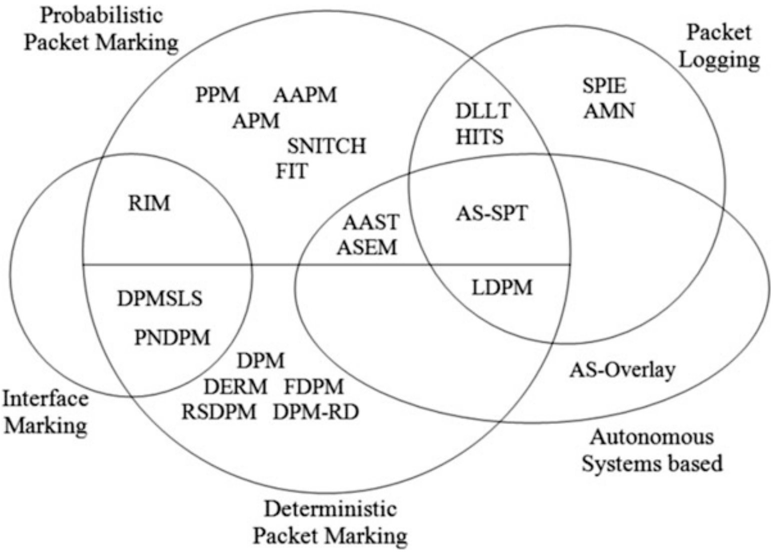


Fig. 7.2 Relation between various traceback mechanisms

7.5 Network Forensic Traceback

Mitropoulos et al. [39] surveyed various approaches for IP traceback and classified them so that the power of digital forensics may be enhanced and the limitations of classic incident handling and response capabilities may be countered. The focus is on their nature (host based, network based, or both), behavior (proactive or reactive), architecture (centralized or distributed), applicability (local network, autonomous systems, or the Internet), and complexity (number of reengineering functions to be performed).

Carrier and Shields [40] propose the Session TOken Protocol (STOP) to assist in the forensic investigation and traceback of a malicious host. The protocol is based on the identification protocol (IDENT) and is aimed to automatically trace attackers logging through a series of stepping stones. STOP saves the user and application-level data associated with a particular TCP connection and returns a random token. It also allows hosts that are not present in the connection chain to make requests on behalf of another host. STOP modifies the request message to provide more options and the response message to protect privacy. The request types allow tokens to be generated along the entire path of hosts. ID request type saves the user name and returns a random token. SV type saves the user name and also the data associated with the process. ID_REC and SV_REC are the recursive daemons which require a random session identifier.

Daniels [41] proposed a functional reference model using passive approach for tracing network traffic. The general reference model for passive origin identification defines the components in terms of their general behavior and goals. Passive approaches do not modify traffic, but they store observations for later analysis. The model has network monitors that communicate with analysis program through a reporting unit. The reporting unit provides the observation information, and a control unit interprets commands from the analysis program. The analysis program can query the monitors for observations and correlate observations to determine the origin of network data elements (packets).

Demir et al. [42] propose two lightweight novel approaches, session-based packet logging (SBL) and SYN-based packet marking (SYNPM), for traceback by providing simple and effective logging. These techniques store log information for longer periods and respect the privacy of communications as well. SBL uses the SYN and FIN packets to record only the critical information (IP addresses and the duration of communication) over the logging period. The header information and the first four bytes of data payload of each logged packets are recorded. SYNPM enables the router to insert distinguishable identifiers in the first SYN packet whenever it routes it. The identifiers are special signatures of routers and are appended to the packets to record the router along the path.

Cohen [43] explores the problem of determining the real source behind the network address translation (NAT) gateway. The author presents a model for disentangling observed traffic into discrete sources and relies on correlation of a number of artifacts which allow the identification of sources. The author based the attribution model on streams defined as a set of packets with the same source and destination addresses and sources which are set of streams attributed to the same host. Assignment of streams to a particular source is handled in an optimized way using energy function. The energy function reduces when a correct assignment is made and increases otherwise. Energy function can be constructed based on attributable artifacts like IP IDs, HTTP referrers, cookies, etc.

A payload attribution system (PAS) is one of the core components in a network forensic system enabling investigation of cybercrimes on the Internet. Ponec et al. [44] proposed several new methods for payload attribution, which utilize Rabin fingerprinting, shingling, and winnowing. The accuracy of attribution increases with the length of the excerpt and the specificity of the query. The collected payload digests can be stored and queries performed by an untrusted party without disclosing any payload information. Guan and Zhang [45] explained the open problems in attack traceback and attribution.

7.6 Conclusion

Attribution in network forensics is a technique which traces the source of the attack packets and supports forensic investigation. A number of IP traceback mechanisms are available which have been discussed. The two approaches of traceback are introduced: reactive and proactive. A review of packet marking techniques, such as probabilistic packet marking (PPM), and deterministic packet marking (DPM), is provided. The chapter explains both the ways and there is always a debate on the choice of these techniques. We suggest using hybrid approach for better support to the forensic investigator. A lot of work has been carried out upon autonomous system-based traceback technique. Router and interface marking technique also describes that to use for attribution. At last, network forensic attribution is explained and its work in practice.

7.7 Questions

Objective Questions

1. Packet traceback can be broadly divided into __________________________ and __________________________.
2. SPIE stands for ____.
3. Traceback was developed as response to mitigate ________________ attacks.
4. Autonomous system number is ______ bit number for IPv4 and ______ bit number for IPv6.
5. Number of interfaces on a standard router are _________.

Short-Answer Questions

1. Compare probabilistic and deterministic packet marking.
2. What is the role of packet marking in network forensics?
3. What additional information can we store in an IPv6 packet for network forensic traceback?
4. What are the limitations in traceback for network forensics?
5. What additional information can be stored about packets which can determine its source?

Long-Answer Questions

1. What are the problems in deploying a traceback mechanism as a solution to limit cybercrime?
2. Compare various traceback mechanisms and illustrate which technique goes closest to the attacker.

References

1. Aljifri H (2003) IP traceback: a new denial-of-service deterrent? IEEE Secur Priv 1(3):24–31
2. Belenky A, Ansari N (2003) On IP traceback. IEEE Commun Mag 41(7):142–153
3. Gao Z, Ansari N (2005) Tracing cyber attacks from the practical perspective. IEEE Commun Mag 43(5):123–131
4. Garfinkel SL (2010) Digital forensics research: the next 10 years. Digit Investig 7(Supplement 1):S64–S73
5. Santhanam L, Kumar A, Agrawal DP (2006) Taxonomy of IP traceback. J Inf Assur Secur 1(1):79–94
6. Stone R (2000) CenterTrack: an IP overlay network for tracking DoS floods. In: Proc. 9th Usenix Security Symposium, Denver, USA
7. Burch H, Cheswick B (2000) Tracing anonymous packets to their approximate source. In: Proc. 14th systems administration conference (LISA 2000), New Orleans, Louisiana, USA, pp 319–327
8. Snoeren AC, Partridge C, Sanchez LA, Jones CE, Tchakountio F, Kent ST, Strayer WT (2001) Hash-based IP traceback. In: Proc. ACM annual conference of the special interest group on data communication (SIGCOMM' 01), San Diego, California, USA, pp 3–14
9. Baba T, Matsuda S (2002) Tracing network attacks to their sources. IEEE Internet Comput 6(2):20–26
10. Zhang L, Guan Y (2006) TOPO: a topology-aware single packet attack traceback scheme. In: Proc. workshop of the 1st international conference on security and privacy for emerging areas in communication networks (SecureComm' 06), pp 1–10
11. Savage S, Wetherall D, Karlin A, Anderson T (2001) Network support for IP traceback. IEEE/ACM Trans Networking 9(3):226–237
12. Song DX, Perrig (2001) Advanced and authenticated marking schemes for IP traceback. In: Proc. twentieth annual joint conference of the IEEE computer and communications societies (INFOCOM 01), Anchorage, Alaska, pp 878–886
13. Dean D, Franklin M, Stubblefield A (2002) An algebraic approach to IP traceback. ACM Trans Inf Syst Secur (TISSEC) 5(2):119–137
14. Aljifri H, Smets M, Pons A (2003) IP traceback using header compression. Comput Secur 22(2):136–151

15. Yaar A, Perrig A, Song D (2005) FIT: fast internet traceback. In: Proc. 24th annual joint conference of the IEEE computer and communications societies (INFOCOM 05), vol 2, Miami, FL, USA, pp 1395–1406
16. Al-Duwairi B, Govindarasu M (2006) Novel hybrid schemes employing packet marking and logging for IP traceback. IEEE Trans Parallel Distrib Syst 17(5):403–418
17. Jing YN, Tu P, Wang XP, Zhang GD (2005) Distributed-log-based scheme for IP traceback. In: Proc. the fifth international conference on computer and information technology (CIT' 05), Shanghai, China, pp 711–715
18. Gong C, Sarac K (2008) A more practical approach for single-packet IP traceback using packet logging and marking. IEEE Trans Parallel Distrib Syst 19(10):1310–1324
19. Jing WX, Lin XY (2009) IP traceback based on deterministic packet marking and logging. In: Proc. eighth international conference on embedded computing, Scalable Computing and Communications (SCALCOM-EMBEDDEDCOM '09), Dalian, China, pp 178–182
20. Bellovin SM, Leech M, Taylor T (2000) ICMP traceback messages. Internet Draft: draft-bellovin-itrace-00. txt 2000
21. Mankin A, Massey D, Chien-Lung W,. Wu SF, Lixia Z (2001) On design and evaluation of "intention-driven" ICMP traceback. In: Proc. tenth international conference on computer communications and networks (ICCCN 01), Arizona, USA, pp 159–165
22. Belenky A, Ansari N (2003) IP traceback with deterministic packet marking. IEEE Commun Lett 7(4):162–164
23. Belenky A, Ansari N (2007) On deterministic packet marking. Comput Netw 51(10):2677–2700
24. Rayanchu S, Barua G (2005) Tracing attackers with deterministic edge router marking (DERM). In: Ghosh R, Mohanty H (eds) Distributed computing and internet technology, vol 3347. Springer, Berlin/Heidelberg, pp 400–409
25. Lin I, Lee TH (2006) Robust and scalable deterministic packet marking scheme for IP traceback. In: Proc. IEEE Global Telecommunications Conference (GLOBECOM '06), San Francisco, California, USA, pp 1–6
26. Jin G, Yang J (2006) Deterministic packet marking based on redundant decomposition for IP traceback. IEEE Commun Lett 10(3):204–206
27. Xiang Y, Zhou W, Guo M (2009) Flexible deterministic packet marking: an IP traceback system to find the real source of attacks. IEEE Trans Parallel Distrib Syst 20(4):567–580
28. Hawkinson J, Bates T (1996) RFC 1930: guidelines for creation, selection, and registration of an Autonomous System (AS). Available: http://tools.ietf.org/html/rfc1930, 30 Apr 2011
29. IANA (2008) 16-bit autonomous system numbers. Available: http://www.iana.org/assignments/as-numbers/as-numbers.xml, 30 Apr 2011
30. Paruchuri V, Durresi A, Kannan R, Iyengar SS (2004) Authenticated autonomous system traceback. In: Proc. 18th international conference on advanced information networking and applications (AINA 04), Fukuoka, Japan, pp 406–413
31. Gao Z, Ansari N (2007) A practical and robust inter-domain marking scheme for IP traceback. Comput Netw 51(3):732–750
32. Udaya Kiran T, Vijay V, Srini Rao P (2009) DoSTRACK: a system for defending against DoS attacks. In: Proc. ACM symposium on applied computing (SAC' 09), Honolulu, Hawaii, pp 47–53
33. Korkmaz T, Gong C, Sarac K, Dykes SG (2007) Single packet IP traceback in AS-level partial deployment scenario. Int J Secur Netw 2(1):95–108
34. Castelucio A, Ziviani A, Salles R (2009) An AS-level overlay network for IP traceback. IEEE Netw 23(1):36–41
35. Chen R, Park JM, Randolph M (2006) RIM: router interface marking for IP traceback. In: Proc. IEEE global telecommunications conference (GLOBECOM '06), San Francisco, California, USA, pp 1–5
36. Chen R, Park JM, Marchany R (2007) A divide-and-conquer strategy for thwarting distributed denial-of-service attacks. IEEE Trans Parallel Distrib Syst 18(5):577–588

37. Yi S, Xinyu Y, Ning L, Yong Q (2006) Deterministic packet marking with link signatures for IP traceback. In: Lipmaa H, Yung M, Lin D (eds) Information security and cryptology, vol 4318. Springer, Berlin/Heidelberg, pp 144–152
38. Peng D, Shi Z, Tao L, Ma W (2007) Enhanced and authenticated deterministic packet marking for IP traceback. In: Xu M, Zhan Y, Cao J, Liu Y (eds) Advanced parallel processing technologies, vol 4847. Springer, Berlin/Heidelberg, pp 508–517
39. Mitropoulos S, Patsos D, Douligeris C (2005) Network forensics: towards a classification of traceback mechanisms. In: Proc. workshop of the 1st international conference on security and privacy for emerging areas in communication networks (SecureComm' 05), Athens, Greece, pp 9–16
40. Carrier B, Shields C (2004) The session token protocol for forensics and traceback. ACM Trans Inf Syst Secur (TISSEC) 7(3):333–362
41. Daniels TE (2004) A functional reference model of passive systems for tracing network traffic. Digit Investig 1(1):69–81
42. Demir O, Ji P, Kim J (2007) Session based packet marking and auditing for network forensics. Int J Digit Evid 6(1):1–15
43. Cohen MI (2009) Source attribution for network address translated forensic captures. Digit Investig 5(3–4):138–145
44. Ponec M, Giura P, Wein J, Bronnimann H (2010) New payload attribution methods for network forensic investigations. ACM Trans Inf Syst Secur (TISSEC) 13(2):1–32
45. Guan Y, Zhang L (2008) Attack traceback and attribution. In: Voeller JG (ed) Wiley handbook of science and technology for homeland ecurity. Wiley, New York

Part III
Advances

Chapter 8
Botnet Forensics

Learning Objectives

- Understanding botnets and the importance of botnet forensics
- Study the botnet forensic process models
- Understanding of various botnet forensic frameworks
- Knowledge of standard tools available for botnet forensics

8.1 Introduction

A botnet is a network of compromised computers controlled by attackers from remote location via C&C channels [1]. The compromised computers are called drones, and the attacker controlling the botnet is called botmaster.

The attackers infect large numbers of vulnerable computers via any primary infection mechanism and guide them to communicate to C&C servers. Further, these infected computers get the secondary payload and other instructions from the C&C servers. In this way attackers remotely control the botnet army and use them for many illegal activities even without owners' knowledge.

These botnets are designed to propagate and communicate in a very covert manner and avoid detection by traditional security tools installed on the network terminals. Botnets run autonomously in a very covert manner and the attacker forwards commands to the bots' army via the randomly compromised hosts in the Internet. This mechanism puts the attacker in the far background and makes it very difficult to trace to the botmaster.

The various botnets discovered in the wild have caused huge financial losses to enterprises, governments, Internet Service Providers, educational systems, and even home users. In June 2014, FBI estimated GameOver Zeus botnets was responsible for more than $100 million loss to many banking and online services

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_8

before its takedown [5]. In the report FBI estimated that 500 million computers are compromised annually, incurring global losses of approximately $110 billion globally [2].

The botnets provide large distributed platforms to perform various malicious activities. Attackers use botnets for a variety of nefarious applications such as launching DDoS attacks, spamming, phishing, spying, click fraud, mining bitcoins, brute force password attacks, and many other malicious activities [3].

The detection of these botnets and traceback to the attackers are very difficult. Moreover, even if attacker location is traced or the C&C servers are located, the cross border presence makes the law enforcement issues very challenging. Therefore, botnet forensics is required to thoroughly analyze the botnets to improve security tools and techniques. The forensic investigation is also needed to collect the evidences to be used to seek permissions to remove the C&C servers and any law enforcement.

The use of computing devices like smartphones, tablets, personal computers, workstations, and other high-end servers connected with the high-speed Internet pave the way for attackers to scan, probe, infect, and highjack these computers to grow their botnet army. Moreover, the malicious content in various forms attract the users to download rogue software, malicious free games, and files and click phishing mails to get them infected.

The botnets run autonomously using multilayer architectures. They communicate using many covert techniques thus changing to track the attackers. The botnets have constantly evolved to more sophisticated and complex structures since the start in 1993. The first-generation botnets used IRC as their C&C protocol. During last few years, botnets using Web servers as C&C channels and HTTP as communication protocol have been discovered. These traditional centralized botnets exhibit C&C traffic leading to be detected, and thus suffer from a single point of failure [4]. Further, the malware authors developed decentralized botnets based on peer to peer (P2P) networks to overcome the weakness of single point of failure. Even after, the hybrid polymorphic botnets are also discovered in the wild. Understanding of the botnet features is important to detect, measure, and compare botnets.

In this chapter we explain the botnet threats to the Internet world. The botnet architectures, protocols, and life cycle are explained for the basic understanding of the threat. We explain the standard botnet forensic process and investigation techniques. Further, we discuss the usefulness of such forensic investigation process against the botnets. Finally, the identified research challenges are explained to serve as the future research direction.

8.2 Botnets Forensics

The first botnet, Eggdrop, was discovered in 1993 being used for sharing user information to protect channels. The botnet was based on IRC protocol. Afterward botnets have constantly evolved to more sophisticated and complex structures. Many small to large malicious botnets have been developed with multiple features and functionalities. Further botnets adopted different architectures and continuously evolved with diverse C&C protocols.

Very few botnets are developed from scratch, while most of the botnets discovered are modified versions of some earlier ones. The first-generation botnets used IRC as their C&C channels. The second-generation botnets evolved using Web servers as C&C and HTTP as communication protocol. Figure 8.1a shows the

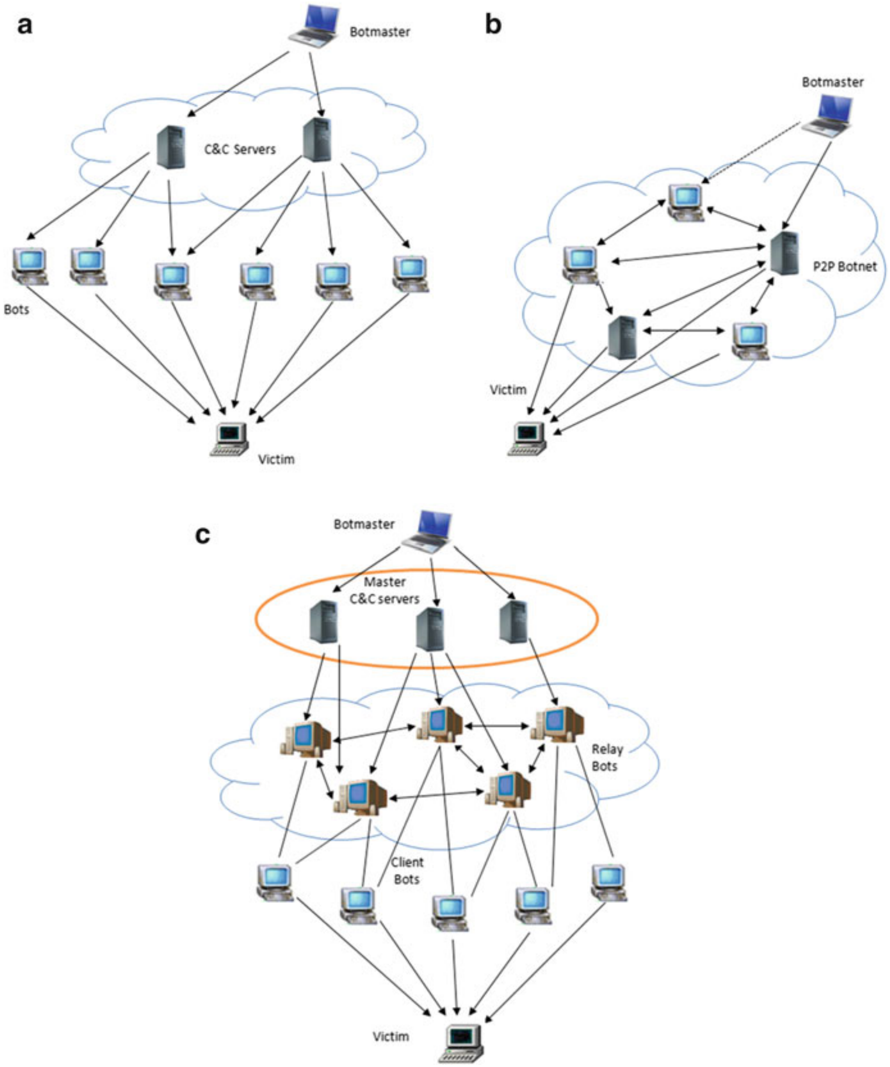


Fig. 8.1 The three types of botnet architectures. (**a**) Centralized (IRC/HTTP) Botnet Model (**b**) Decentralized P2P Botnet Model (**c**) Hybrid Botnet Model

architecture of the centralized botnets. These centralized botnets exhibit similarity in the C&C traffic leading to expose the servers, and thus suffer from a single point of failure [4]. The increased rate of botnet detections propelled the malware authors to develop decentralized botnets to overcome the weakness of single point of failure. Figure 8.1b shows topology of such decentralized P2P botnet model. These distributed botnets are based on the peer to peer (P2P) networks. The distributed P2P architecture leads to complicated network and non-efficient control. So, the hybrid botnets with the desirable features of both centralized and distributed architecture emerged. Figure 8.1c shows the topology of hybrid decentralized P2P botnet model with super peers.

The botnet uses various covert techniques during the different phases of its life cycle [5]. Botnets exploit many integral flaws, vulnerabilities, and social engineering techniques. Everyday numbers of vulnerabilities are exploited in different applications to infect the computers with various malware.

Botnets are differentiated by the C&C of the architecture. The C&C component of botnet architecture is used to control the bots from remote system. This forms the multi-tier architecture of botnets and differentiates them from other malwares. Botnet master compromises systems to set up the C&C servers to issue commands and get the results back from the *bots*. The botnets are usually classified according to their C&C architecture [6, 7]. Over the course of time bot authors have developed C&C channels based on different network structures. Cooke et al. [8] proposed three different botnet communication topologies: centralized (IRC based and HTTP based), decentralized (P2P based), and random.

The botnets employ multilayer architecture which keeps the attacker in the far background and thus makes it difficult to reach to him [9]. In the partially decentralized layered P2P botnets, some peers are controlled by the attacker to issue and disseminate commands/information to other peer bots. The P2P botnets take advantage of the flexible self-organizing network infrastructure, and the peer bots are easily added and replaceable with other hosts.

Botnets can propagate or communicate using either push-based methods or pull-based methods. (1) The push-based methods employ network scanning techniques to find the vulnerable hosts and infect them to turn into a bot. This method reflects the automatic self-propagating nature of the botnets. Conficker [9, 10] and Simda are the well-known examples of this kind of botnets. The push-based method is an active method as it automatically scans and infects new machines to grow the number of victims. (2) In pull-based methods bots propagate with the help of users or other methods, i.e., nonself-propagating methods. In this method bot masters compromise Web servers, upload the malicious codes, and drive users to download the malicious codes (social engineering). These methods also distribute the bot codes via already-infected machines using pay-per-install (PPI) scheme. Unlike the push-based methods, the pull-based methods are passive as these involve human actions or other malware to infect new machines. MegaD and Srizbi botnets employ pull-based method. Recent botnets use pull-based methodology to infect new machines [11]. This shift of botnet propagation from push to pull has made prevention and investigation more difficult. Such botnets are called downloader botnets. BredoLab and Botsniffer are some the examples such botnets.

Botnet forensics is a science that exhibits the process of getting bot clues in order to identify, acquire, analyze, attribute, and mitigate the botnet threats. Botnet forensics gets the root cause of security breaches. This may start from simple evidence for DDoS attack, phishing emails to other malicious information. The forensics provides important insight on how botnets behave in the wild. Further the information obtained may be useful to even identify new instances of botnets. The investigation may take place from the services run by host and suspected or vulnerable activities.

The botnet detection, investigation, and tracking require a forensic environment. The forensic environment uses software tools and techniques to collect digital evidence for understanding how botnets propagate, attack, or perform malice. The information obtained from the forensic investigation is useful for thorough understanding of botnet features and firepower. Further, the information may be used to detect C&C servers and track the attacker. Unfortunately, attackers use anti-forensic techniques and remain anonymous by using number of proxies.

The botnet forensic system is a security device with hardware and pre-installed software. There is a requirement of integrating existing knowledge of botnets and using the live forensic and tracking techniques. Live forensics is the process of imaging-infected machine, documenting the steps when it is running, and collecting all identified evidences without any changes.

Since botnets are the root cause of most of the cyber attacks and threats, the collaborative efforts are required between security firms, IPSs, DNS providers, research groups, and the law enforcement agencies. Further, educating users against social engineering tricks, improving system resilience, and hardening the computer systems are required to prevent and mitigate the effects of such threats. The centralized botnets employ a dedicated C&C infrastructure as a main source or sink of commands and data. Thus, it can be easily spotted and taken down. The decentralized P2P botnets distribute the command channel across multiple infected peers to be more resilient and thus preludes the traditional takedown techniques [12], [13].

The botnet structures have grown manifold in the past years, further, using advanced packing and encryption techniques; therefore the analysis is very challenging. Furthermore, even after detection the takedown operations/mechanisms involve different parties, e.g., registrars or hosting providers. Moreover, the global spread of the botnet needs the permission of various law enforcement agencies, who need correct evidences about the botnet for the required takedown and mitigation operation. Therefore, though botnet forensics is very difficult it is a pressing need against the botnets disrupting the Internet-based economy and the related privacy concerns.

8.3 Acquisition

The data acquisition is the first and important phase in botnet defense. Data acquisition is also part of the botnet detection. Acquisition can be performed at the host and network level. The host-level data acquisition captures the data, logs, and other details from the operating system. The network-level data acquisition may be performed at network interface card or routers level.

The botnet acquisition may involve data collection from the malicious operations and the botnet binary collection for further analysis and investigation. Honeypots are generally used to collect the malware binaries and log the operations. Further, the captured malware samples can be run in the sandboxes to analyze the botnet behavior, tools, and techniques.

Holz et al. [14] introduced a methodology to track botnets. The authors emphasized that in order to track the botnets, some information is required to be gathered by the honeypots. Further, based on this work Cremonini and Riccardi designed a Dorothy framework [15] to monitor the activities of the botnet named as siwa. The authors infiltrated and monitored a botnet to collect the information about the structure, communication, command language, distribution, and functions of the botnet.

The dynamic botnet analysis approaches also employ the well-established honeypot techniques. Honeypot is defined, “An information system resource whose value lies in the illicit use of the resource [16].” In this technique system vulnerabilities are exposed to the attackers and let the systems get infected/compromised and then capture the botnet binary, monitor the operations the malware performs, and log the traffic or information generated. This method also protects actual production systems by sinkholing the attacks/compromise.

The effective deployment of the honeypots forms a collection known as the “honeynet.” The honeynet systems collect the suspicious scanning or probing traffic. The deployed honeypots may have low- or high-interaction level to the botnet. The low-interaction honeypot: HoneyD [17]. The high-interaction honeypot is a separate computer system that logs any suspicious activity and also mimics to perform as directed by the botnet master monitoring the applications running on the host and the ports they are communicating. Further, examining the TCP stream of any suspicious connections may help to discover the IP address of the C&C server. The content of the communication can be analyzed to get the details.

Cavalca and Goldoni [18] proposed Honeynet Infrastructure in Virtualized Environment (HIVE), an automated malware collection and analysis architecture. The authors used the architecture to collect various botnet malware to form the botnet code database which is useful/requirement to investigation in the botnet forensics. Further, they suggested possibilities of using the analysis services from the external providers.

Honeypots are successfully deployed in botnet defense and investigation system. Unfortunately, attackers counter all defense measures and develop tools and techniques to evade detection and stay hidden to continue the operations. The advanced botnets have the features to stay stealthy, hidden, and even detect any virtual environment.

Zou and Cunningham [19] presented the honeypot detection method based on the assumption that honeypots deployed for security operation cannot be allowed to participate in the actual malice operation. Attackers detect the honeypot based on the trial whether the compromised host can successfully relay attack commands. These honeypot-aware botnets detect the virtual environments like honeypots and sandboxes before they perform any malice activities. If such environment is detected, botnets exit and/or provide fake information.

Sandbox The execution of the botnet code can present information about the botnet infiltration into the system and the malicious activities like disk read/write and the network communication to get commands, download updates, or other binaries and spamming, etc. The botnet binaries can be run in a controlled environment and monitored in order to log the actions performed by the botnet. The commands executed and the external hosts contacted are monitored, and further analysis is done go get more insight of botnet.

8.4 Analysis

The botnet forensics includes the analysis of various components of the botnets including C&C servers/channels and compromised host. Further, forensic analysis process includes analysis of bots' functionality, C&C servers/traffic, botnet attack, and botnet design.

The previous attacking nature and behavior of botnet helps to identify the intention and method of the attacker. For this purpose analysis is required to obtain the exact facts from the evidence, so that it can mitigate. Analysis classifies and correlates the whole incident into different groups according to their behavior and nomenclature either for mitigating the effect of crime or permanently sorting out the crime. This analysis can be done through the different data mining soft computing tools or different machine learning techniques. TCPDump, Wireshark, TCPFlow, TCPTrace, OllyDbg, IDA Pro, NetFlow, TCPXtract, Snort, etc., are the different tools for supporting the botnet attack analysis.

- Analysis by modeling the botnets
- Analysis of the real botnets captured in the wild

The various botnet forensic analysis techniques can be studied into two broad categories: static and dynamic.

Static Analysis It is the method of understanding the malware behavior without executing it. The method includes analysis of log files, analysis of file systems, and the suspects of malware presence. To gain more insight of the malware, internal structure reverse engineering is performed and has no potential threats to the environment.

Dynamic Analysis This approach is performed by executing the botnet binary in a controlled environment, e.g., a sandbox. This approach studies and monitors the external view, thus, analog to the black-box testing approach. It mainly monitors the behavior and operations of the botnet. This is complementary to the static analysis and performance. The objectives of the approach are to understand the functions and features of the botnet. Often, this is performed in a virtual environment and controls the various parameters to study the behavior at different circumstances. Unfortunately, some botnets are even coded to recognize the virtual environment and thus exit immediately and clear the logs or provide false information.

Botnets can be detected by analyzing their flow characteristics. Several strategies have been proposed to dynamically analyze and defeat botnets. Barford and Yegneswaran [20] performed an in-depth analysis of the malicious bot source programs. The authors found that the botnet architecture and implementation are complex. The authors discovered that the Agobot employs test for debuggers and VMware, kills antiviruses, or provides false information.

Botnet analysis is to enumerate the bots in the botnet. Rossow et al. [13] proposed a graph-based P2P botnet model to capture the properties and vulnerabilities of the botnet. The authors also analyzed the resilience of the botnets to the takedown efforts. Further, they proposed two P2P node enumeration techniques, crawling and sensor injection, for the botnet size measurement. The P2P botnets are susceptive to command injection attacks. Sality botnet employs a peer reputation scheme. The current P2P botnets are quite resilient to disruption attacks.

There is some possibilities through reverse engineering the domain generation algorithm (DGA) for registering the domain prior in which the bot can communicate at any future point which helps to ignore accessing to bot herder, and under their own control, it requests. It gives the different perspective among infected host and botmaster.

Holz et al. [14] proposed the concept for analyzing and monitoring the bot through honeypot technologies. This work gave the immense motivation to the development of Dorothy framework [15]. This framework provides very automated features to analyze, track, and visualize a botnet. Such framework was also made for the requirement of finance-based botnet investigation [21].

Botnet analysis includes two major ways to analyze the malware: examining the code and behavior. The botnet always tries to evade detection and also makes the bots code analysis harder.

(a) *Static Code Analysis:* There are many tools available for the code analysis. Some tools are simple, while others require significant efforts by the investigator, e.g., Hex editor WinHex (a popular tool for static analysis). The investigators require a copy of the malware code for analysis. The malware code may be in the form of a script or compiled binary code. The analysis of the simple script files is easier, while binary compiled files need binary decompiler routine to understand/determine the features and functioning. Unfortunately, uses of some decompilers require strong program understanding, and further, some botnet binaries prevent the use of decompilers. The botnet authors also use packers to minimize the size of the bot binary code, obfuscate binary data, and limit the function of the decompiler. Further, the combination of packing utility with encryption makes reverse engineering more difficult, as well as the effectiveness of the unpacking or decryption tools. The use of uncommon packing utilities even makes the unpacking task more difficult.

The static analysis tools identify the run time errors and security vulnerabilities. The tools also provide the valuable insight and information such as symbol tables, parse trees, and call graphs valuable for botnet analysis.

(b) *Run Time Code Analysis:* Static code analysis is not a complete solution to get botnet evidence. Sometimes static examination of bot binary is unable to analyze bot. To evade from the detection of existence, binary itself uses the different packets and encryption methodology. it may be necessary to execute the malware to monitor the actions that take place on the victim system, such as file system changes, registry changes, and network activity for gathering additional information of malware behavior. Run time code analysis is particularly helpful to identify such botnet network information where is the location of the bot connection to receive commands, as well as any usernames, handles, IRC channels, and passwords.

An important step of forensic investigation is preserving the evidence of any cybercrime for prosecution. The malware analysis is a considered to be the main activity of digital forensics. The analysis can further classified into following ways:

Spam-Based Analysis

Compromised system is that which used to send spam messages. The advantage of performing spam using botnet has reliance as even if we can identify a bot sending and are able to block it, still there will be other bots that will still be performing spam. For example, Rustock botnet was used for spamming and was sending 25000 messages per host per hour. So the magnitude of spam is huge when using a botnet.

Pathak et al. [22] did a comprehensive study of content agnostic characteristics of spam campaign. During the collection and analysis, non-proxy spamming domains were observed to exhibit spamming duration far longer than a five-day period whose effect was studied on spam campaign signature generation. Further analysis revealed workload distribution, sending patterns, and coordination among the spamming machines.

Pitsillidis et al. [23] described the mechanism for better filtering of spam by analyzing the vantage points of a spammer. By monitoring botnet host, we are able to identify new spam as it is created, and later we can create proper strategies to deal with such spams. This technique gave precise decisions with no false positive.

Distributed Denial of Service (DDoS)-Based Analysis

Freiling et al. [24] describe a prevention mechanism, which operates by infiltrating and analyzing mechanism of the remote controlling the bots. The method can be used over the Internet and infiltrate mostly IRC bots which is the most common type of botnet architecture used by botmasters.

Thomas et al. [25] explore the Koobface zombie infrastructure and analyzes its effect. It was discovered that despite domain blacklisting service by social network, over 213000 users were compromised, generating over 157000 clicks.

Provos et al. [16] presented a state of *malware and Web and emphasized* on the installation of malware which can be easily installed by analyzing the vulnerabilities of the host once it clicks on the malicious link.

Fast Flux-Based Analysis

Passerini et al. [26] developed a system named FluXOR to detect and monitor fast flux service networks. The detection totally relies on the analysis of a set of features observable from a view of a victim of a scam. Nazario et al. [27] established the fact that active lifetime of fast flux botnet varies from less than one day to months. The domains used for fast flux are registered months before they are used and kept as dormant.

The authors in [28] performed detailed technical analysis of the Festi botnet and discovered the distinguishing features of the botnet. The botnet implement object-oriented architecture into the kernel-mode driver to make it portable. The botnet also exhibits strong resistance to forensic analysis and has the ability to bypass IDS/IPS software tools.

Traffic-Based Analysis

In the Internet, we receive the data in the form of packets. The number of packets we send and receive from the Internet is called network traffic. Network traffic exhibits the packets we receive and sends to the destination. Broadly the traffic can be categorized in two ways: simple traffic and active traffic.

Simple Traffic In simple traffic, the timely delivery and quality of services are confirmed and given priority. The source of the traffic has expectation to deliver the packets in time. The traffic is also known as sensitive traffic. The examples include VoIP, video conferencing, online gaming and Web browsing, etc.

Active Traffic In this category the traffic ensures the quality of services with the speed. Active traffic is not sensitive to quality of service metrics such as jitter, packet loss, latency, etc. Sometimes we receive unwanted traffic which carries worms, botnet, or malicious activities. The Internet is full of malicious activities such as phishing, Denial of Services, click fraud, spamming, *etc., so therefore we* need to analyze the botnet traffic.

Botnet traffic is a process of generating, recording, reviewing, and analyzing the botnet traffic for the purpose of acquiring, identifying, detecting, and mitigating the botnet attacks. This is the process of using manual and automated technique to review cluster detail and statistics within botnet traffic. Botnet traffic analysis is done through bandwidth monitoring software tools. Traffic statistics refer and help in following:

- Understanding the botnet and utility
- Evaluating the botnet
- Downloading and uploading the speeds
- The content, size, source, and destination of the information
- Identifying malicious and suspicious packets, etc.

(i) *Command-and-control-based traffic analysis*

Masud et al. [29] propose a temporal correlation technique for C&C traffic detection. Using the temporal correlation of two host-based log files, the author-illustrated bots react faster than the human operators. The authors applied this technique in log files for detecting the bot activity in a system using TCPDump and exedump. This tool records inflow and outflow network traffic packet and the start time of the application execution at the host machine, respectively. The authors apply data mining to extract relevant features from the log files and detect the C&C traffic.

(ii) *P2P-based traffic analysis*

By analyzing Waledac botnet, Dae-il et al. [30] proposed their study on infected HTTP2P botnet and also facilitated their detection. In order to breach the network security, infected botnet changed the protocol. In the beginning botnet used only IRC protocol. The botnet suggested by Dae-il et al. utilized multiple protocols which include TCP,UDP, HTTP, and so on. The infected bots utilize combination of protocols for instance in case of HTTP2P, i.e., P2P over HTTP. In case of HTTP2P, HTTP protocol enjoys the merits being firewall friendly, whereas P2P protocol is helpful in evading a client and server architecture. This prompted the author for proceeding further toward analyzing the characteristics of the phases of botnet behavior communicating and utilizing the HTTP2P protocol. For the study the author classified Waledac botnet into two categories, they were proxybot and workbot. The study results facilitated the detection of HTTP2P botnet in the network traffic.

Dafan et al. [31] studied the different phases in between the general peer to peer protocol and advance peer to peer protocol. The attacker hardcodes a search key in their bot program, which looks for the order command for future attack with the search key on regular time intervals. On the grounds of unstructured peer to peer protocols, the author has designed a very upgraded hybrid peer to peer botnet. He mentioned the different requirements for the peer to peer protocol and showed general peer to peer protocol does not require global information.

(iii) *IRC-based traffic analysis*

Mazzariello et al. [32] addressed centralized botnet detection. C&C structure provides the simplicity to create attacking scenario by the bot herder. Once C&C channel is identified, the whole botnet can be dismantled. He experimented and found that the known bots are characterized by the propagation mechanism. This tendency comes after inheriting the same strategies and characterizing by the next bot from the popular bot.

Karasaridis et al. [33] designed to measure the distance between monitored flow data and predefined IRC traffic flow.

(iv) *Flow-based traffic*

Shahrestani et al. [34] analyze the current network intrusion detection method. This method depends upon anomaly detection and passed from the flow-based botnet detection system to check trustfulness. Through visualization, it is then aggregated to reveal malicious traffic. Finally this information is forwarded for validation.

(v) *DNS network traffic*

Thomas et al. [35] designed for DNS-based detection. He described and analyzed the tracking and analysis for P2P version-2. His experiment captured result based on DNS and data hash list size. After maintaining large hash lists, the results explained the ability of TRAPP-2 to detect traffic under a saturated network load. They analyzed the DNS traffic to identify the malware family without the need for obtaining malware sample. He made the cluster of DNS traffic through different infected machine.

8.5 Attribution

To understand the functioning of botnet is important for attribution of the botnet. A typical botnet has several components and follows a different file cycle. Botnet forensics is forced to show the different communication paths between an infected system and initial point of attack origin. Botnet forensics uses incident response, prosecution, and the tough measurement of the botnet to identify the attacker. The attacker uses different techniques such as IP spoofing and stepping stone attack to hide himself. For this purpose attribution is required to find out the evidence and the kind of attack.

Attribution shows collection of the comprehensive information of botnet samples, further deploys honeypots, and traces the details about the developers.

Botnet attribution is done for various operations and purposes such as botnet size measurement, nodes classification and locating C&C servers, and even tracing botmasters. Rossow et al. [13] investigated botnet for its size measurement. The botnet investigation may start with collection of samples or passive monitoring of bots behavior.

The botnet attribution processes are the study of propagation, C&C structure, communications protocols, attacks, and victim investigation. Botmasters may also periodically perform the query to check DNS Blacklists (DNSBL) to check if their bots are listed in the blacklist. Ramachandran et al. [36] proposed a passive analysis of the DNS monitoring activity of the attackers to detect the bot nodes.

The botnet attribution can be performed in two major ways. (a) Capture the botnet binaries, collect the real botnets traffic in the wild, and perform various analysis methods to understand the botnet. (b) Run the known/unknown botnet binary in a controlled environment to monitor the behavior and log its traffic. The investigator is able to control all the variables in a controlled environment, but, this leads to a trade-off between the level of control and the realistic behavior of the malware. Such experiments help to dissect the behavior of the botnet.

An investigation of BredoLab botnet by National High Tech Crime Unit of the Netherlands' Police Agency (NHTCU) in 2010 estimated the three million infected machines and then on October 25, 2010 got access to hosting server in the Netherlands and successfully took over the botnet.

Cusack (2014) [11] concluded that botnet investigation is a complex process, and controlling the cost of botnet investigation is critical. Therefore, technical processes are required to be automated and to control the time and cost resource. He emphasized that the success of the integrated investigation framework will depend on the comprehensive centralized database maintained by the stakeholders.

Botnet attribution is a complex operation and requires significant time and efforts. Therefore, expert knowledge, automation, collaboration, and sharing tools and techniques are required for effective investigation and analysis.

Botnet attribution covers all the components of the botnet including the victim investigation, infected host investigation, network traffic, and C&C server.

The C&C server is the most important component in botnet which may serve in many ways in different botnets. The C&C server may host commands, spam templates, stolen email ids, etc.

Network-Based Attribution

Network-based attribution is primarily based on traffic monitoring, detecting C&C servers, phishing Web sites, and analysis of traffic. This attribution is the botnet attribution specific on IRC, HTTP, DNS, P2P, mobile, cloud, etc.

Host-Based Attribution

However, many researchers proposed botnet attribution at the network level, but botnet attribution at the infected host level is also important and has potential advantages. Practically, both approaches are relevant and in fact complementary to each other.

Law et al. [37] used the host-based botnet investigation approach, and further, the authors highlighted that the host-based approaches are easier than network-based approaches requiring large traffic log storage and flow monitoring as the required data collation is less.

Ard et al. [38] defined botnet attribution's two phases as: (1) The analysis of the malware shows the postmortem of binaries. This kind of attribution is also known as run time analysis for identifying the network and its information. (2) The second phase of attribution is tracking sources for identifying the DNS name registers, the IRC controllers, and servers. The authors presented the study of basic analysis techniques for reverse engineering botnets. Some P2P botnets also use custom protocols which make reverse engineering to get an insight of the botnet [13].

Cusack [11] proposed the botnet forensic techniques. The authors applied the various proposed investigation guidelines and concluded that it is possible and feasible to investigate botnet attacks, but controlling the cost of investigation is critical. The author further recommended quantifying botnet investigations into five levels of cost: based on time, complexity, and technical requirements.

Table 8.1 illustrates the botnet forensic tools used at the data collection and analysis phases.

Obstacles in Botnet Attribution

Botmaster performs recon and even anti-recon actions to continue their illegal services. They even may perform DNSBL reconnaissance queries to check if their services are blocked [36]. Botmasters also implement anti-recon strategies in response to the takedown attempts by reconnaissance. The anti-recon strategies may include attacks such as DDoS against recon nodes, automatic black listing nd reputation schemes [12].

Different security agencies such as FBI and Microsoft have taken step and experimented to reduce Zeus botnet family's threat.

Botnet forensics follows that the strategy of prevention is better than attack for its defensive approach. Botnet forensics provides and spends attacker's most of the time and energy to maintain their route through trapping network so that he can get the least time to launch the attack. These restrictions improve the security issues and reduce the crime rate. Thus criminal cannot harm or penetrate the real network very easily. In order to ensure the network security and to fix the accountability, the

Table 8.1 The tools used for botnet forensics

Process	Tools	Purpose
Malware collection	Dionaea	A low-interaction honeypot that collects a copy of the malware exploiting vulnerabilities
Virtualization	VMware workstation, Oracle Virtual Box	Tools for visualizing the computer system
Forensic Image	Helix Pro	A forensic tool for incident response
Memory analysis	Volatility framework	A forensic tool that can extract various types of information from a memory image
Initial virus scan	Virus total	A public service that analyzes suspicious files and URLs
Initial sandbox analysis	Anubis, CWSandbox	Public services that analyze the behavior of Windows PE executable with special focus on the analysis of malware
Packer detectors	PEiD v 0.94	A tool for detection of packers, cryptors, and compilers for Windows PE executable
String extractor	BinText v3.03	This tool search ASCII, Unicode, and resource strings in a file
Dissemblers and debuggers	IDA Pro, OllyDbg	Reverse engineering tool

Internet service providers are made responsible for the activities on their networks. Further in this direction one more significant step is taken, now it is mandatory for the firms engaged in e-commerce and online business to reveal their security breaches and disclosure of majors taken to ensure the network security as a part of statuary compliance so that the scale of harm, damage or loss it caused, share of bandwidth used, and traffic load on the network can be minimized.

Botnet measurement is useful to calculate the size of the botnet, estimate the growth and predict the future changes in the botnet model. The results of botnet measurement help to design better prediction or defense systems against present and future botnet threats.

The forensic attribution is very important to better understand the ever evolving botnet picture and threat. Botnet developers use continuously evolving technology to increase the stealth and covertness of the botnets. Therefore, united efforts are required to mitigate the botnets at the successive phases.

Botnet developers use the full potential/advantage of interconnectivity of the Internet to infect and undertake large numbers of computers from all over the world. Unfortunately, this is a major hurdle in botnet defense and investigation.

Botnet forensic is very essential to gather the information about botnets and analyze how they propagate, behave along protection and detection systems, and perform malicious actions. The thorough understating of botnets is further useful to develop the more effective protection systems against botnet threats.

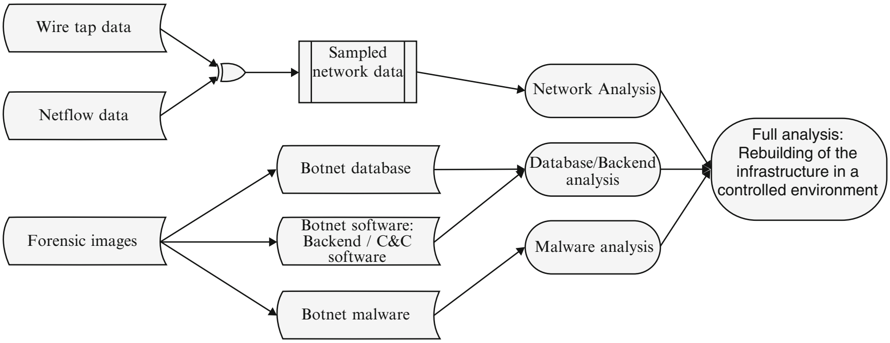

Wire tap data
Netflow data
Sampled network data
Network Analysis
Botnet database
Database/Backend analysis
Forensic images
Botnet software: Backend / C&C software
Malware analysis
Botnet malware
Full analysis: Rebuilding of the infrastructure in a controlled environment

Fig. 8.2 Botnet forensic attribution model [39]

Botnet forensic framework is inclusive of botnet identification, bot classification, bot analysis, and identification of potential bot attacks. The success of any botnet attack attribution requires the effective extraction of features, behavior, signature, and protocol from the different networks.

The risks and challenges are involved while executing the malware binaries in the controlled environment. In the system/process of the risk management, one key functionality is to protect the integrity of the data collected/evidence from the infected system or suspected traffic and to ensure the security of the investigator information system/tools.

Graaf et al. [39] proposed a botnet forensic investigation model to investigate and analyze botnet called BredoLab (first discovered in 2009). The authors identified that the traditional forensic investigation models were not much effective to investigate and analyze the large BredoLab botnet resources. BredoLab serves as large malware downloading platform and is also used for installing malware to third parties, i.e., used for pay-per-install (PPI). He performed the forensic investigation of the BredoLab botnet and discovered that its infrastructure consisted of C&C servers, a central proxy server, several proxies installed on bots, database server, a back-end server, a personal hacking server, and a VPN server (Fig. 8.2).

8.6 Research Challenges

The analysis of the botnets is not an easy task due to the hybrid and ever-changing nature of botnets. There are many challenges and issues with botnet forensic analysis and investigation. The identified challenges are:

- The botnets continuously advance in the propagation techniques including push based, pull based, and drive-by download to defeat to botnet prevention measures.
- Some bots are even preprogrammed to destroy them if suspected to be caught by defenders and clear the evidences. The bot authors also use many anti-forensic

techniques to challenge the analysis and investigation processes. The botnets even clear the logs once the task is over or suspect the presence of any virtual machine/honeypot.
- The various reports disclose that the botnets adapt in response to the defense deployed by the security fronts and further explore new technologies such as mobile computing, cloud computing, and even Internet of Things (IoT) and mobile botnets [40].
- A proper forensic environment is required to collect data and analyze it. Researchers and security persons use honeynets to attract the botnet compromise and log the tools and techniques used by the botnets. Unfortunately, bot authors use anti-honeypot technique to ignore the honeynets.
- The botnet authors use covert communication by hiding the traffic from the victim users and even the networks administrators.
- None of the studies purely focused on the botnet forensics and discovered the anti-forensic techniques. Therefore, the future research would also focus on developing research methods for botnet investigation which are more effective, direct, and systematic.

8.7 Conclusion

The botnets are the main source of large number of cyber threats and attacks. Botnets cause huge financial, social, and privacy damage all across the world. To combat the botnet threats, integrated and shared repositories are required to be built for standard references. Some countries and banking association are working to build the blacklist of infected IPs. The chapter explains the effectiveness of various botnet forensic acquisition, analysis, and attribution techniques.

Tackling botnets requires a collaborative effort between researchers, ISPs and DNS providers, law enforcer, and self-organized security communities such as shadow server. The chapter may be useful to design standard botnet forensic tools and conduct the botnet investigation research in a systematic way. Further, it provides guidelines for hardening the computing system, educating users, and for improving the resilience of the systems to further attack.

8.8 Questions

Multiple Choice Questions

1. The phase is not part of the botnet forensic framework.

   (a) Investigation
   (b) Identification
   (c) Analysis
   (d) None of the above

2. In botnet forensic attribution model cover.

   (a) Malware analysis
   (b) Attribution
   (c) Acquisition
   (d) None of the above

3. Analysis of botnet forensics not covered.

   (a) Static analysis
   (b) Run time code
   (c) Dynamic analysis
   (d) None of the above

4. Which of the following is not a part of the botnet life cycle?

   (a) Infection
   (b) Invocation
   (c) Communication
   (d) Attack

5. Which of the following statement is incorrect?

   (a) IRC-based botnets are prone to detection due to centralized server.
   (b) HTTP-based botnets are prone to detection due to centralized server.
   (c) P2P-based botnets are prone to detection due to centralized server.
   (d) None of the above

6. Fast-flux mechanism is used by botnets authors for ________

   (a) Ensure the availability botnet service
   (b) Make the botnets more resilient
   (c) Both A and B options
   (d) None of the above options

7. Which of the following is/are used for data acquisition?

   (a) Honeypots
   (b) Sandboxes
   (c) Both (A) and (B) options
   (d) None of these options

8. Which of the following is part of the botnet forensic framework?

   (a) Investigation
   (b) Identification
   (c) Analysis
   (d) All of the above

9. The botnet forensic attribution model covers ____________
   (a) Malware analysis
   (b) Attribution
   (c) Acquisition
   (d) None of the above
10. Analysis of botnet forensics do not cover ___________
   (a) Static analysis
   (b) Run time code
   (c) Dynamic analysis
   (d) None of the above

Short-Answer Type Questions

1. What is botnet?
2. What are the various threats posed by botnets?
3. Explain the significance of botnet forensics.
4. Elaborate different phases in investigation of botnet attacks.
5. What is difference between run time code and static code analysis?

Long-Answer Type Questions

1. Explain the different botnet architectures.
2. Discuss the requirement of botnet forensics.
3. Explain the process of data acquisition step of botnet forensics.
4. Discuss the botnet process model in details.
5. What are the various tools used botnet forensics. Explain in details.

References

1. Wang P et al (2010) Honeypot detection in advanced botnet attacks. Int J Inf Comput Secur (IJICS) 4(1):30–51
2. Stevenson A (2014) Botnets infecting 18 systems per second, warns FBI. July 16, 2014 [cited 2015 9 March 2015]; Available from: http://www.v3.co.uk/v3-uk/news/2355596/botnets-infecting-18-systems-per-second-warns-fbi, 31 Mar 2016
3. Rajab MA et al (2006) A multifaceted approach to understanding the botnet phenomenon. In: Proceedings of the 6th ACM SIGCOMM conference on internet measurement (IMC'06), ACM, Rio de Janeiro, Brazil
4. Grizzard JB et al (2007) Peer-to-peer botnets: overview and case study. In: Proceedings of first workshop on hot topics in understanding botnets (HotBots'07), USENIX Association, Cambridge, MA, pp 1–8
5. Rodríguez-Gómez RA, Maciá-Fernández G, García-Teodoro P (2013) Survey and taxonomy of botnet research through life-cycle. ACM Comput Surv (CSUR) 45(4):45
6. Zhu Z et al (2008) Botnet Research Survey. In: 32nd annual IEEE international computer software and applications (COMPSAC'08)
7. Feily M, Shahrestani A, Ramadass S (2009) A survey of botnet and botnet detection. In: Third international conference on emerging security information, systems and technologies (SECURWARE'09). IEEE

8. Cooke E, Jahanian F, McPherson D (2005) The Zombie roundup: understanding, detecting, and disrupting botnets. In: Proceedings of the USENIX workshop on steps to reducing unwanted traffic on the internet (SRUTI ’05). Boston: USENIX Association, Berkeley, CA
9. Seungwon S et al (2012) A large-scale empirical study of conficker. IEEE Trans Inf Forensics Secur 7(2):676–690
10. Fitzgibbon N, Wood M (2009) Conficker. C: a technical analysis. SophosLabs, Sophon Inc
11. Cusack B (2014) Botnet forensic investigation techniques and cost evaluation. In: Proceedings of the conference on digital forensics, security and law
12. Andriesse D, Rossow C, Bos H (2015) Reliable Recon in adversarial peer-to-peer botnets
13. Rossow C et al (2013) SoK: P2PWNED – modeling and evaluating the resilience of peer-to-peer botnets. In: IEEE symposium on security and privacy (SP)
14. Bacher P et al (2005) Know your enemy: tracking botnets. In: The Honeynet Project & Research Alliance
15. Cremonini M, Riccardi M (2009) The Dorothy project: an open botnet analysis framework for automatic tracking and activity visualization. In: European conference on computer network defense (EC2ND)
16. Provos N, Holz T (2007) Virtual honeypots: from botnet tracking to intrusion detection. Addison-Wesley Professional
17. Provos N (2003) Honeyd-a virtual honeypot daemon. In: 10th DFN-CERT workshop, Hamburg, Germany
18. An open architecture for distributed malware collection and analysis. (2010)
19. Zou CC, Cunningham R (2006) Honeypot-Aware advanced botnet construction and maintenance. In: International conference on dependable systems and networks (DSN ’06)
20. Barford P, Yegneswaran V (2007) An inside look at botnets. In: Christodorescu M et al (eds) Malware detection- advances in information security. Springer US, pp 171–191
21. Riccardi M et al (2010) A framework for financial botnet analysis. In: eCrime Researchers Summit (eCrime), 2010
22. Pathak A et al (2009) Botnet spam campaigns can be long lasting: evidence, implications, and analysis. ACM
23. Pitsillidis A et al. Botnet judo: fighting spam with itself
24. Freiling F, Holz T, Wicherski G (2005) Botnet tracking: exploring a root-cause methodology to prevent distributed denial-of-service attacks. Computer Securityâ€“ESORICS 2005, pp 319–335
25. Thomas K, Nicol DM. The Koobface botnet and the rise of social malware. IEEE
26. Passerini E et al (2008) Fluxor: detecting and monitoring fast-flux service networks. In: Detection of intrusions and Malware, and vulnerability assessment (DIMVA’08), Lecture Notes in Computer Science
27. Nazario J, Holz T (2008) As the net churns: fast-flux botnet observations. In: 3rd international conference on Malicious and unwanted software (MALWARE ’08), Alexandria, VA
28. Matrosov A, Rodionov E (2011) Festi botnet analysis & investigation
29. Masud MM et al (2008) Flow-based identification of botnet traffic by mining multiple log files. IEEE.
30. Dae-il J et al (2009) Analysis of HTTP2P botnet: case study waledac. In: Communications (MICC), 2009 IEEE 9th Malaysia International conference on
31. Dafan D et al (2008) Deep analysis of intending peer-to-peer botnet. In: Grid and cooperative computing, 2008. GCC ’08. Seventh international conference on
32. Mazzariello C (2008) IRC traffic analysis for botnet detection. Ieee
33. Karasaridis A, Rexroad B, Hoeflin D (2007) Wide-scale botnet detection and characterization. In: Proceedings of the first conference on first workshop on hot topics in understanding botnets. Cambridge, MA
34. Shahrestani A et al (2009) Architecture for applying data mining and visualization on network flow for botnet traffic detection. In: Computer technology and development, 2009. ICCTD ’09. International conference on

35. Thomas B et al (2011) An FPGA system for detecting malicious DNS network traffic advances in digital forensics VII. Springer, Boston, pp 195–207
36. Ramachandran A, Feamster N, Dagon D (2006) Revealing botnet membership using DNSBL counter-intelligence. In: Proceedings of the 2nd workshop on steps to reducing unwanted traffic on the internet (SRUTI'06), San Jose, California, USA
37. Law FYW et al (2010) A host-based approach to BotNet investigation? In: Goel S et al (eds) Digital forensics and cyber crime. Springer, Berlin/Heidelberg, pp 161–170
38. Ard C (2007) Botnet analysis. Int J Forensic Comput Sci 2(1):65–74
39. de Graaf D, Shosha A, Gladyshev P (2013) BREDOLAB: shopping in the cybercrime underworld. In: Rogers M, Seigfried-Spellar K (eds) Digital forensics and cyber crime. Springer, Berlin/Heidelberg, pp 302–313
40. Vural I et al (2010) Mobile botnet detection using network forensics. In: Future internet – FIS. Springer, Berlin/Heidelberg, pp 57–67

Chapter 9
Smartphone Forensics

Learning Objectives

- Understanding the importance of smartphone forensics
- Study of smartphone forensic process models
- Discussion on various smartphone forensic frameworks
- Knowledge of standard tools available for smartphone forensics

9.1 Introduction

Smartphones play a vital role in everyone's life these days. A smartphone device is like a friend in which the user stores personal information. A smartphone enhanced with a hardware and software capability not only serves as a means of communication but also as a small-scale portable computer with advanced communication capability. This explosive growth of smartphones has drawn the attention of cybercriminals who try to trick the user into installing malicious software on the device. Through these malicious software, the attacker can steal the user's private information from the devices. Broadly, there are three locations, device, network, and data center, where the attackers may exploit vulnerabilities to launch malicious attacks. Many security solutions are given by the researchers to provide the security on smartphones. However, these security solutions can also be taken by criminals to hide their criminal activities which they performed through smartphones. Recently, in Columbus, Ohio, Detective Zane Kirby, a forensic examiner for the Franklin County Internet Crimes against Children Task Force, helped to convict a 23-year-old man accused of trying to solicit an inappropriate relationship with a 13-year-old girl. The case breaker was lewd photos of the perpetrator found on his phone and sent to the victim, originally recovered from the suspect's phone, together with deleted call logs and the girl's name listed in the suspect's contact list. The usage of smartphone data provided strong enough evidence to result in a guilty verdict for the defendant.

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_9

In National Institute of Standards and Technology Guidelines on Cell Phone Forensics [1], smartphone forensics is defined as “the process of recovering digital evidence from a smartphone device under forensically sound conditions and utilizing acceptable methods.” Forensically sound is a term used extensively in the digital forensic community to qualify and, in some cases, to justify the use of a particular forensic technology or methodology. Indeed, many practitioners use the term when describing the capabilities of a specific software or when describing a forensic analysis approach [2].

According to Casey [3], smartphones are the cell phone with PDA functionalities as they used to make calls like cell phones and for transferring the data like images, audio, and video over the Internet like PDA devices via multiple links, e.g., UMTS, WLAN, Bluetooth, and IR, which conclude that smartphones store crucial information of user. These potential evidences are placed in the smartphones increase the necessity of smartphone forensics for the forensic investigator in the field of digital forensics. Bennett [4] also mentioned that the use of smartphone in online transactions such as stock trades, flight reservations and check-in, smartphone banking, and communications regarding illegal activities that are being utilized by criminals has created a need for smartphone forensics.

The possible evidence items are subscriber and equipment identifiers, date/time, language, and other settings, phonebook/contact information, calendar information, text messages, call logs, electronic mail, photos, audio and video recordings, multimedia messages, instant messaging, Web browsing activities, electronic documents, social media-related data, application-related data, location information, geolocation data, etc.

According to Al-Zarouni [5], smartphones are always active and are constantly updating data, which can cause faster loss of evidentiary data. Second, the operating systems (OS) of smartphones are generally closed sources, with the notable exception of Linux-based smartphones, which makes creating custom tools to retrieve evidence a difficult task for forensic examiners. In addition, smartphone vendors tend to release OS updates very often, making it hard for forensic examiners to keep up with the examination methods and tools required to forensically examine each release. The variety of proprietary hardware of smartphones is another issue faced by forensic examiners. Additionally, Lessard et al. [6] mentioned that one of the major difficulties in the field of smartphone forensics is the general lack of hardware, software, and/or interface standardization within the industry. This fact makes forensic processing a hard task, especially for unified research. All these facts described above make a forensic processing a hard task.

Smartphone security has become increasingly important as it relates to the security of personal and business information now stored on these devices. Now, smartphones not only serves as a means of communication but also as a means of planning and organizing their work and private life. Within companies, these technologies are causing profound changes in the organization of information systems, and therefore they have become the source of new risks. Indeed, smartphones collect and compile an increasing amount of sensitive information to which access must be controlled to protect the privacy of the user and the intellectual property of the company. The attackers may exploit vulnerabilities in the smartphones at the location, device, network, and data center. Different security counter measures are being developed and applied to smartphones to keep the device and user information safe. But when it comes to smartphone forensics, these security measures applied on devices work as anti-forensic techniques in the forensic investigation. This becomes a research challenge for the smartphone forensic investigator.

Before starting any forensic investigation, the investigators and the forensic team need to follow some principles. The Association of Chief Police Officers (ACPO) [7] suggested the four principles when dealing with digital evidence:

- No actions performed by investigators or their agents should change data contained on digital devices or storage media.
- In exceptional cases, individuals accessing original data must be competent to do so and be able to explain their actions.
- An audit trail or other record of all applied processes must be created and preserved for an independent third-party review.
- The person in charge of the investigation has overall responsibility for ensuring the abovementioned procedures and principles are followed.

The first principle is however not possible with smartphone since the phone has to be kept switched on in the order to acquire data from it. Switching on the phone or connecting the phone to a computer will very likely change some data, even without explicitly doing so. This means that in the best case, data must be modified as little as possible.

9.2 Smartphone Forensic Process Models

A process model is a defined standard or method of getting things done by applying scientific methods. Smartphones are manufactured by different companies and each company defines its own standard platform and storage system. Hence there is no standard and universally accepted process model for smartphone forensics. Based on the operating system used in the smartphone, some of the process models are discussed:

(i) Symbian phone process model.

Yu et al. [8] developed an adaptive process model based on different versions of Symbian smartphones. The model contained the different stages of forensics shown in Fig. 9.1.

The preparation and version identification phase are divided into two parts: one for accessing official public information on the target Symbian smartphone and other was to make ready, Symbian evidence of tools and accessories of evidence. The initial version of the information and tools that are used to identify and

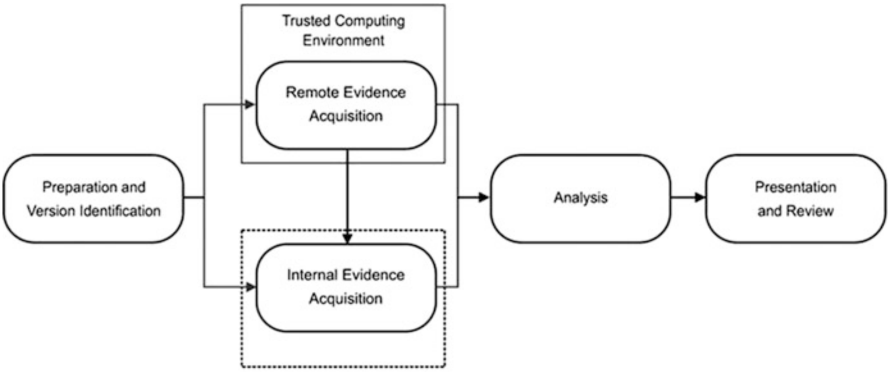

Trusted Computing Environment
Remote Evidence Acquisition
Preparation and Version Identification
Internal Evidence Acquisition
Analysis
Presentation and Review

Fig. 9.1 Process model for Symbian phone

prejudge the credibility of the smartphone, working in the TCE (trusted computing environment). Next, the remote evidence acquisition phase has two methods, protocol approach and hardware approach, so that it can acquire the evidence from the device. Protocol approach is based on command-response protocols by connecting to a remote host computer, such as AT command set, SyncML, OBEX, and Nokia FBUS proprietary protocol [9]. Hardware approach permits to acquire a binary image file of the entire flash memory content. The JTAG debug port can access the data within the flash memory [10, 11]. In the internal evidence acquisition phase, investigators can access the entire memory through the acquisition tools [12] and then copy the files to removable media. Analysis phase used different methods to analyze retrieved data like static code analysis, including pattern matching, lexical analysis, abstract syntax tree analysis, data flow analysis, and other methods. The last presentation and review phase presented the result of acquired and analyzed data. The result report involved reviewing all the steps in the investigation process and identifying areas of improvement. After this the results and their subsequent interpretations can be used for the examination and analysis of evidence in future investigations.

(ii) Windows phone process model

Anup Ramabhadran [13] proposed the Windows based smartphone forensic process model in order to overcome the major shortcomings of the existing digital forensic model. The model consists of twelve stages as shown in the Fig. 9.2.

The preparation phase involves getting an initial understanding of the nature of the crime and activities like preparing the tools required for standard portable electronic device investigations, building an appropriate team, assigning roles to each personnel, accumulating materials for packing evidence sources, etc. Securing the scene deals with the unauthorized access and preserving the evidence from being contaminated. This phase plays a major role in the overall investigative process as it

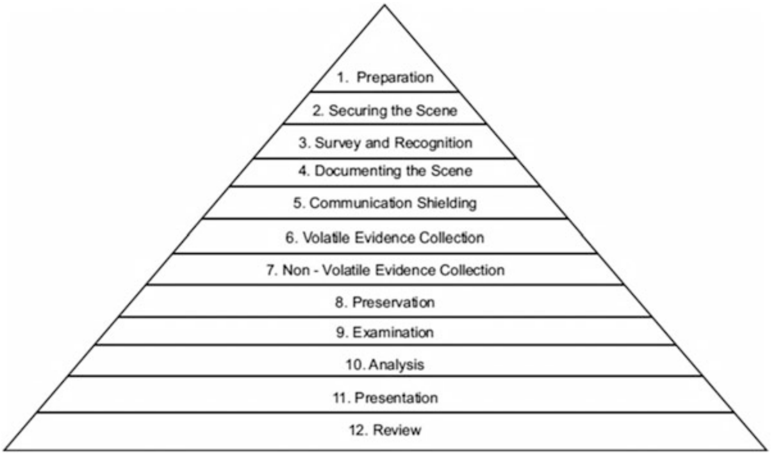


Fig. 9.2 Process model for windows smartphone device

determines the quality of evidence. Survey and recognition phase involves an initial survey conducted by the investigators for evaluating the scene, identifying potential sources of evidence, and formulating an appropriate search plan. Document the scene involves proper documentation of the crime scene along with photographing, sketching and crime scene mapping. The communication shielding phase blocked all further possible communication options of the devices. The volatile evidence phase collects the volatile evidence which is present in ROM. Non-volatile evidence collection involves collecting evidence from external storage media supported by these devices, like MMC cards, compact flash (CF) cards, memory sticks, secure digital (SD) cards, USB memory sticks, etc. Further the preservation, examination, analysis, presentation, and review are same as in the existing process models. The model is applicable to corporate and law enforcement investigations and incident response activities alike.

(iii) Android phone process model

Aleksandar et al. [14] proposed the harmonized digital forensic investigation (HDFI) process model. The proposed model is a multi-tiered model which compromises the twelve phases shown in Fig. 9.3, and each phase contains a set of subphases. Stacey et al. [15] did the forensic investigation on Android phone at which a phishing attack is launched via SMS, to check and verify the workability of harmonized digital forensic process model for smartphone forensics. The investigation proved that the HDFI process model successfully incorporated an Android device.

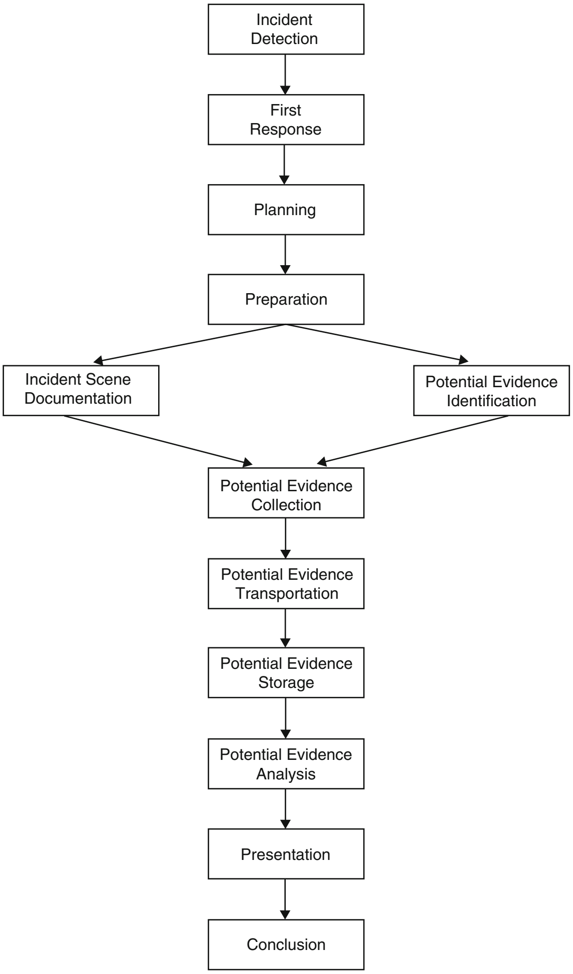

Incident Detection
First Response
Planning
Preparation
Incident Scene Documentation
Potential Evidence Identification
Potential Evidence Collection
Potential Evidence Transportation
Potential Evidence Storage
Potential Evidence Analysis
Presentation
Conclusion

Fig. 9.3 Harmonized digital forensic investigation process model

9.3 Standard Process Model

Although there is no universally accepted standard process model for smartphone forensics, there are certain sequence of actions (acquisition, examination, analysis, and reporting of retrieved data) defined by Jansen and Ayers [1] which are important to investigate any smartphone in forensically sound manner. Afterward, Rick et al. [16], in the revised addition of NIST documentation, stated that a forensic investigation must be a sequence of actions, consisting of the following steps:

- Preservation
- Data Acquisition
- Examination and analysis
- Reporting

Preservation

Preservation involves the search, recognition, documentation, and collection of electronic-based evidence. In order to use evidence successfully, whether in a court of law or a less formal proceeding, it must be preserved. Failure to preserve evidence in its original state could jeopardize an entire investigation, potentially losing valuable case-related information. The first responders who first arrive at the scene start the preservation phase. The responders' task is to secure and cordon off the scene and ensure the security of all individuals. Next, the entire scene is documented using camera/video so that the permanent record of the scene is created. The team then determines whether there is a need for any kind of DNA analysis to be conducted. But then some challenges which need to be handled during this phase are as follows:

- If the phone found is damaged.
- If the phone is not damaged, then the state of phone, i.e., on/off state.
- If the phone is on, then there is a need to isolate a phone network.

The processing of the above are extremely critical for forensic investigators as a minor mistake may lead to loss of crucial evidence.

Data Acquisition

Broadly there are two methods of forensic acquisition: logical and physical acquisition. Casey [3] mentioned that logical acquisition retrieves a bitwise copy of entities such as a files and directories that reside inside a logical storage. Grispos et al. [17], Quick et al. [18], and Thing et al. [19] defined the physical acquisition as a bitwise copy of the internal flash memory. There is a one more acquisition method called manual acquisition defined as whatever an individual is capable of acquiring by interacting with the device itself, but this method has high probability of human errors, so this is used as a supplementary with the other two methods named above. The data acquisition type that is selected by a forensic examiner is determined by a number of factors, including time constraints, what data acquisition types a particular model of handset is supported for, if only live data is required, if deleted data is required, if third-party application data is required (e.g., WhatsApp), the skill level and experience of the analyst, and the tools available to them. Additionally Rick et al. [16] mentioned the various types of acquisition tools in five different levels shown in Fig. 9.4. The classification system provides a framework for forensic examiners to compare the extraction methods used by different tools to acquire data. The objective of the tool classification system is to enable an examiner to easily classify and compare the extraction method of different tools.

Manual Extraction: Reviewing the phone documentation and browsing using the device buttons to view and record data by hand. The method is fast and requires

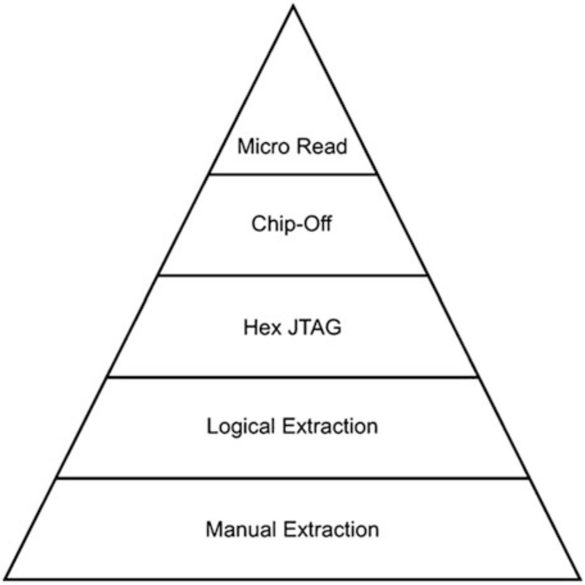


Fig. 9.4 Smartphone device tool classification system

no cable but will not get all data and take more time to complete the process. Acquisition of deleted data is impossible at this level.

Logical Extraction: Connect the data cable to the handset and extract data using commands in client-server architecture. The method is fast but may change data, log file access, and not acquire deleted files.

Hex Dumping/JTAG (Joint Test Action Group): This method pushes the boot loader into phone and dump memory. The method acquires hidden data and deleted data and can even do password bypassing but requires custom cables. Parsing and decoding of the captured data are a major challenge with these methods.

Chip Off: The process of removing the needed chip off the phone's mother board. Chip off is done by physically removing the NAND logic gate flash chip and reading the NAND memory with the NAND reader. This data acquisition method extracts all data from the device memory but difficult to use and may damage chip.

Microread: This method will not be used very frequently as the tools are very expensive, time-consuming, and may damage the device completely. In this a high-power microscope is used to view the state of memory.

Alghafli et al. [20] showed the comparative study of each method in Table 9.1. Each method is different from the other, for example, deleted data could not be acquired through logical acquisition, but the physical acquisition could be applied by the forensic investigators to acquire deleted data.

Table 9.1 Comparison between data acquisition method

Item	Manual	Logical	Physical	Chip-off
Time	Fast	Fast	Slow	Slow
Cost	Cheap	Cheap	Medium	High
OS dependence	Yes	Yes	No	No
Data structure	Simple	Simple	Binary copy of the memory chip without data structure	Binary copy of the memory chip without data structure
Preserve integrity	No	No	Sometimes	Yes
Bypass security code	No	No	Yes	Yes
Works with damage phones	No	No	Yes	Yes
Retrieves deleted files	No	No	Yes	Yes
Retrieves volatile data	Yes	Yes	Yes	No
State of the device	On	On	On and Off	Off

Examination and Analysis

The examination process uncovers digital evidence, including that which may be hidden or obscured. The results are gained through applying established scientifically based methods and should describe the content and state of the data fully, including the source and the potential significance. Data reduction, separating relevant from irrelevant information, occurs once the data is exposed. The examination process begins with a copy of the evidence acquired from the smartphone device. The analysis process differs from examination in that it looks at the results of the examination for its direct significance and probative value to the case. Examination is a technical process that is the province of a forensic specialist. However, analysis may be done by roles other than a specialist, such as the investigator or the forensic examiner. Examination and analysis using third-party tools are generally accomplished by importing a generated smartphone device memory dump into a smartphone forensic tool that supports third-party smartphone device images.

Reporting

Reporting is the process of preparing a detailed summary of all the steps taken and conclusions reached in the investigation of a case. Reporting depends on maintaining a careful record of all actions and observations, describing the results of tests and examinations, and explaining the inferences drawn from the data. A good report relies on solid documentation, notes, photographs, and tool-generated content. Reporting occurs once the data has been thoroughly searched and relevant items bookmarked. Many forensic tools come with a built-in reporting facility that usually follows predefined templates and may allow customization of the report structure.

9.4 Frameworks

Smartphone forensic frameworks include implementation of a solution on various types of models. Some of the recent work has been discussed: Satheesh KS et al. [21] proposed an agent-based approach for database acquisition in Blackberry devices shown in the Fig. 9.5. The proposed approach used a client-server architecture where the agent acts as the server and the client acts as a desktop PC. An agent is a .cod file (BackberryImager.cod) that was developed in the JDE environment. To start data acquisition, the agent was uploaded on the device through Javaloader.exe which is a part of Blackberry SDK. Once the agent is uploaded, the agent started transferring of databases' file from server to client through IChannelEvents. The files are stored in the form of image file at the client side. The authors also developed a complete forensic tool called BAAT, which contains two modules, one is for agent-based acquisition and the other is for analysis. The developed tool has two main advantages, one can read phone information like IMEI number, device ID, and OS version, and the other can fetch databases like Bluetooth log information, Blackberry Messenger, etc. But there is a disadvantage in the developed tool that it cannot access SMS databases.

Park et al. [22] proposed a new analysis technique for fragmented flash memory pages in smartphones. The proposed technique is suitable for both Android and iOS devices. First, the forensic examiner verifies the status of the target device.

If the device is damaged, the data can be dumped by the chip off process of the flash memory. If the device is not damaged, the examiner should judge whether the device complies with the JTAG standard. If JTAG testing is possible, the flash memory connected to the processor can be dumped as an image. Otherwise, the examiner can utilize the backup feature or image the flash memory depending on whether the device can be rooted and the purpose of forensic analysis. The examiner analyzes the files acquired by the backup or debugging feature; if a dump image of the flash memory is acquired, a more systematic analysis can be performed. If it is necessary to reconstruct the flash memory image because the target device is not a block device, such as the eMMC (embedded multimedia card), that integrates the flash memory and its flash controller, the examiner can attempt the reconstruction using the metapage and the spare area. If the reconstruction is successful, file system forensic analysis can be performed. The whole procedure is shown in the Fig. 9.6.

Rizwan et al. [23] implemented an efficient generalized forensic framework for acquisition and subsequent analysis of Android smartphones shown in the

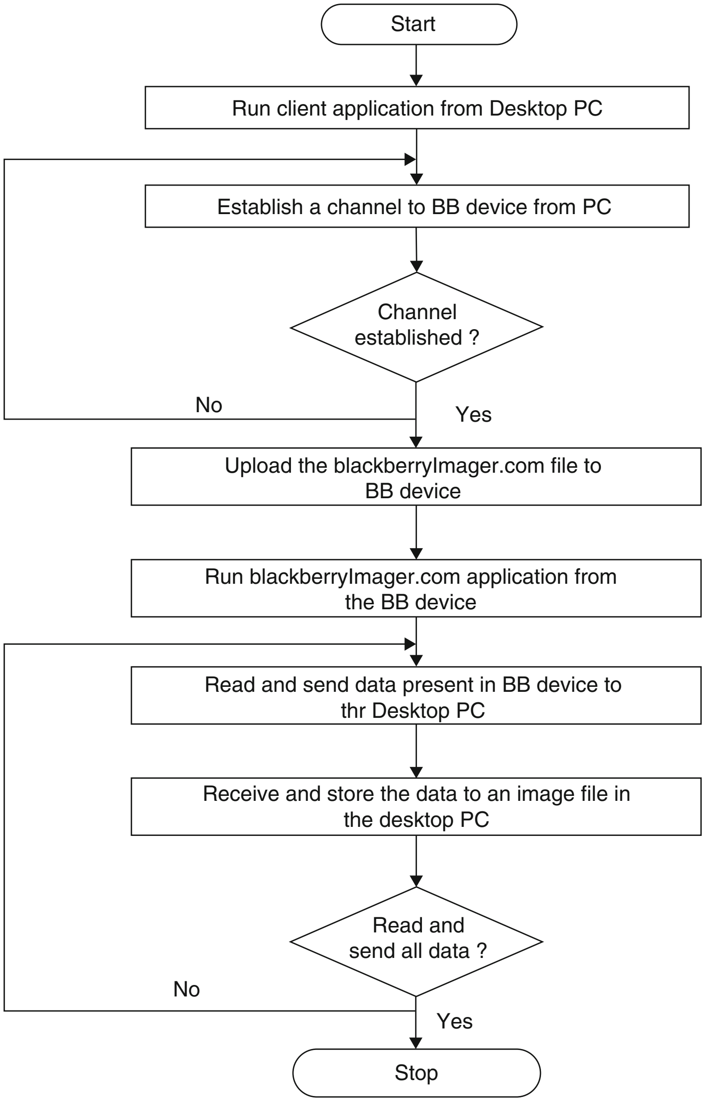
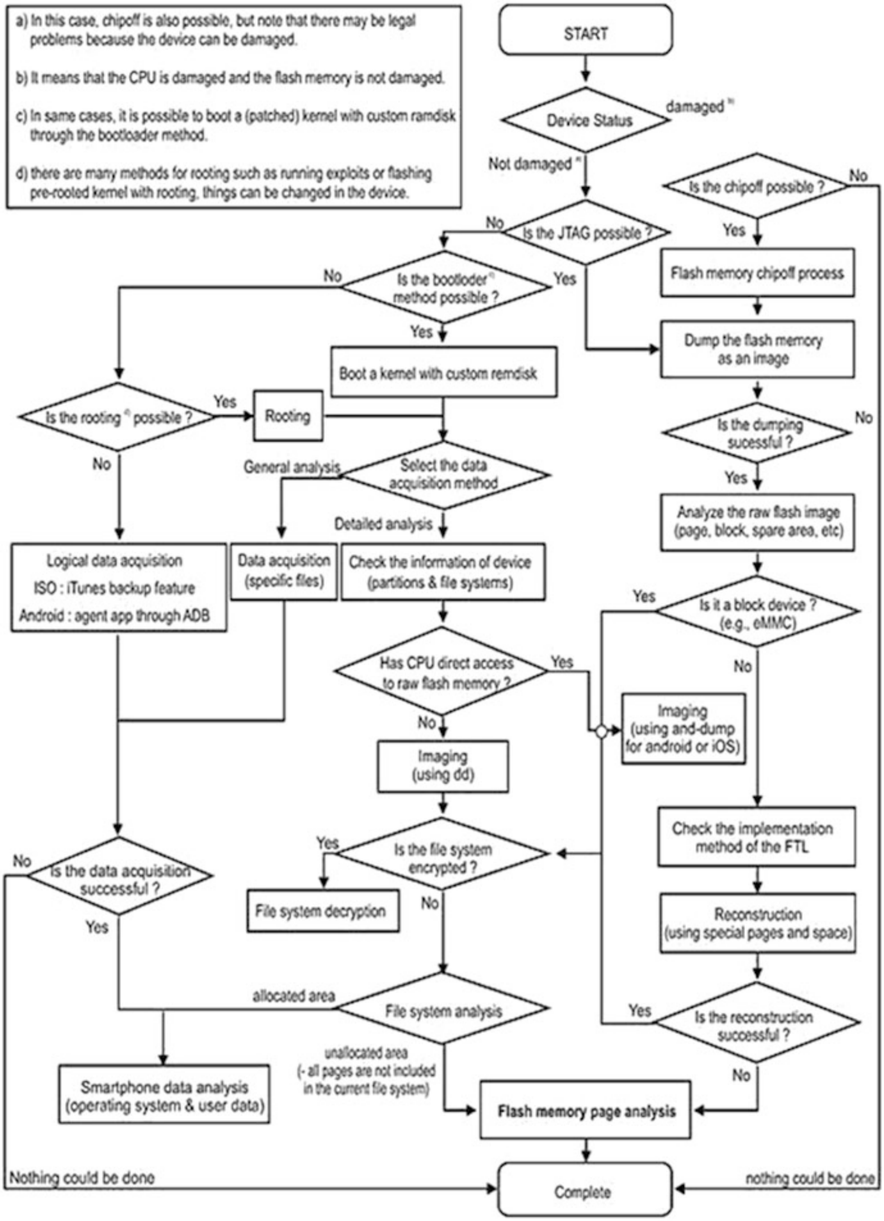

Start
Run client application from Desktop PC
Establish a channel to BB device from PC
Channel established ?
No
Yes
Upload the blackberryImager.com file to BB device
Run blackberryImager.com application from the BB device
Read and send data present in BB device to thr Desktop PC
Receive and store the data to an image file in the desktop PC
Read and send all data ?
No
Yes
Stop

Fig. 9.5 Agent-based acquisition process

Fig. 9.7. First the authors shielded the device with Faraday cage to avoid network communication. The SD card is checked whether it is already plugged in or not.

If the card is plugged in, then replace it with the forensic SD card which contains updated version of Efficient Generalized Forensics Framework Acquisition App. Launch the app through file explorer. The app killed all unnecessary processes running on the system in order to avoid locking problems. To ensure integrity,



a) In this case, chipoff is also possible, but note that there may be legal problems because the device can be damaged.

b) It means that the CPU is damaged and the flash memory is not damaged.

c) In same cases, it is possible to boot a (patched) kernel with custom ramdisk through the bootloader method.

d) there are many methods for rooting such as running exploits or flashing pre-rooted kernel with rooting, things can be changed in the device.

START
Device Status
damaged b)
Not damaged a)
Is the JTAG possible ?
No
Yes
Is the chipoff possible ?
No
Yes
Flash memory chipoff process
Dump the flash memory as an image
Is the dumping sucessful ?
No
Yes
Analyze the raw flash image (page, block, spare area, etc)
Is the bootloder c) method possible ?
No
Yes
Boot a kernel with custom ramdisk
Is the rooting d) possible ?
Yes
No
Rooting
Select the data acquisition method
General analysis
Detailed analysis
Logical data acquisition
ISO : iTunes backup feature
Android : agent app through ADB
Data acquisition (specific files)
Check the information of device (partitions & file systems)
Is it a block device ? (e.g., eMMC)
Yes
No
Has CPU direct access to raw flash memory ?
Yes
No
Imaging (using and-dump for android or iOS)
Imaging (using dd)
Check the implementation method of the FTL
Reconstruction (using special pages and space)
Is the reconstruction successful ?
Yes
No
Is the file system encrypted ?
Yes
No
File system decryption
Is the data acquisition successful ?
No
Yes
allocated area
File system analysis
unallocated area (- all pages are not included in the current file system)
Smartphone data analysis (operating system & user data)
Flash memory page analysis
Nothing could be done
Complete
nothing could be done

Fig. 9.6 Framework for smartphone forensic analysis

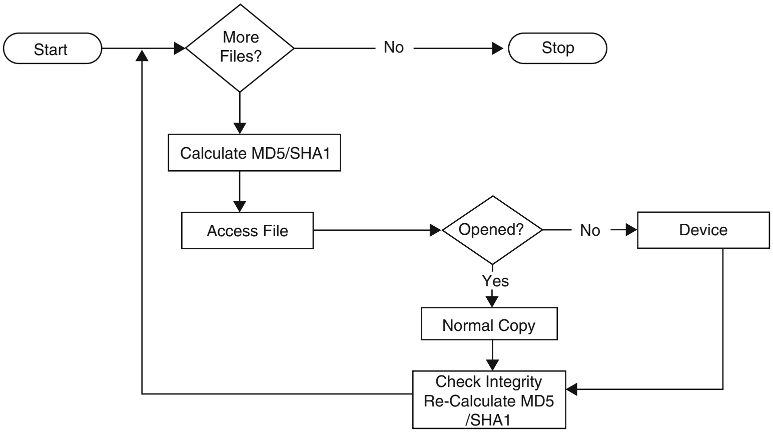


Fig. 9.7 Efficient generalized forensic framework for acquisition

hashing of each file is performed. The relevant information about all the file hashes is saved in Checksum.xml log file for further analysis later.

9.5 Tools

Smartphone forensic tools are still in the development stages. To investigate the smartphones which are involved in a crime or other incident, forensic investigators require tools that allow them the faster examination of information present on the device. A number of existing commercial off the shelf (COTS) and open-source products provide forensic specialists with such capabilities. These tools are required to meet the digital evidence rules so that the integrity of data on the device is maintained. The scenarios serve as a baseline for determining a tool's capability to acquire and examine various types of known data, allowing a broad and probing perspective on the state of the art of present day forensic tools to be made. Following are the list of some commonly used tools by the investigator:

- Cellebrite – Cellebrite leads the smartphone forensic industry with its range of smartphone forensic products, Universal Forensic Extraction Device (UFED) series, providing multiple platform solutions for extraction, decoding, analysis, and reporting of data and passwords from thousands of smartphone devices. UFED solutions are available in a range of platforms to suit the investigation environment and case requirements [24].
- Oxygen Forensic Suite – This is the smartphone forensic software that goes beyond standard logical analysis of cell phones, smartphones, and PDAs. Using advanced proprietary protocols permits Oxygen Forensic Suite and the Analysts Version which are used to extract much more data than usually extracted by

logical forensic tools, especially from smartphones, such as the iPhone and Android [25].

- Device Seizure – This is a comprehensive handheld forensic analysis tool supporting over 1000+ phones. In addition to both CDMA and GSM phone support, Device Seizure's support for PDA systems includes Palm DD Command Line Acquisition (PDD) and PDAs using the following operating systems: Palm through 6, Windows CE/Pocket PC/Smartphone 4.x and earlier, BlackBerry 4.x and earlier, and Symbian 6.0. The Device Seizure software is sold independently or with a cable and device toolbox. Individual cables and connections can be purchased from the manufacturer [26].
- Access Data Mobile Phone Examiner Plus (MPE+): It is a stand-alone smartphone device investigation solution that includes enhanced smart device acquisition and analysis capabilities. With a different approach to digital smartphone forensics, MPE+ allows smartphone forensic examiners to take control of the investigation by providing them with unique tools necessary to quickly collect, easily identify, and effectively obtain the key data [27].
- Now Secure – In 2014 viaForensics became Now Secure. This company has developed many excellent smartphone forensic tools, many of which are free to use. One of the key products is viaExtract, a program which allows the user to extract data from Android devices, crack passphrases, and PINS and to examine images from external (SD) and internal (eMMC) storage cards. This program, which is one of their commercial products, works on many of the most popular Android smartphones and smartphone devices [28].
- Final Mobile – Final mobile Forensics offers one of the most advanced and easy-to-use data carving tools for smartphone forensic community. It captures/analyzes the smartphone device's raw data and uses a database wizard to streamline the acquisition procedure, providing greater acquisition of "deleted" and "live" data that can be undetected with a logical file acquisition. The tool enables investigators to efficiently perform critical tasks during the investigation of cellular phones [29].
- XRY – XRY is a software application tool developed by Micro Systemation which is designed to run on the Windows operating system allowing a secure forensic extraction of data from a wide variety of smartphone devices, such as smartphones, GPS navigation units, 3G modems, portable music players, and the latest tablet processors such as the iPad. XRY comes in XRY software, XRY hardware, and XRY complete. XRY complete is the all-in-one smartphone forensic system combining both our logical and physical solutions into one package. XRY complete allows investigators full access to all the possible methods to recover data from a smartphone device [30].
- Lantern – This is the most cost-effective and comprehensive Mac-based tool for the iPhone, iPod Touch, and the iPad. Providing a complete logical acquisition of the iOS device, Lantern provides examiners with a dedicated solution and a complete reporting tool to showcase the extensive data found on the devices. The latest version, Lantern 2, enables examiners to delve deep into the devices'

database and file system, to retrieve data not acquirable by the average forensic tool [31].

- Secure View – The popular DataPilot phone editing tool, which offers the most extensive collection of tested and supported phones (currently numbering over 1000), has been engineered for digital examiners. In its first iteration, Secure View provides unlimited reads of phones and employs write blocking to secure the data during interrogation. The software is sold alone, or as part of a complete kit, including the DataPilot Universal Cable system. Individual cables can also be purchased from the manufacturer [32].
- CellDEK – This advanced cell phone data extraction device is a self-contained system that features a touch screen display allowing the user to quickly identify devices by brand, model number, dimensions, and/or photographs [33].
- MOBILedit – With MOBILedit Forensic, an investigator can view, search, or retrieve all data from a phone with only a few clicks. This data includes call history, phonebook, text messages, multimedia messages, files, calendars, notes, reminders, and application data such as Skype, Dropbox, Evernote, etc. It will also retrieve all phone information such as IMEI, operating systems, firmware including SIM details (IMSI), ICCID, and location area information. Where possible MOBILedit Forensic is also able to retrieve deleted data from phones and bypass the passcode, PIN, and phone backup encryption [34], (Table 9.2).

9.6 Research Challenges

Smartphone forensics is a discipline which presents a steady growth. The research conducted and undergoing standardization attempts the development in this area.

- There should be accessibility of features and parts of the smartphone devices that are crucial for forensic investigations, such as boot loaders and RAM heap. A boot loader might lack the functionality to copy memory, while the RAM heap might be practically inaccessible.
- Brand and Model Diversity. The great majority of experiments take place on specific brands of smartphone devices and versions of operating systems. It is generally accepted though that even devices that run the same OS present different behavior. This means that while an Smartphone forensics method may be operational and useful for a certain version of a given smartphone platform, it can become obsolete very quickly due, say, to the installation of an OS patch.
- It appears that due to the nature of smartphone, many smartphone forensic procedures must inevitably involve live forensics as the device needs to be powered on (traditional dead forensics are almost useless). There is an upcoming trend of comparison between live and dead forensic techniques. As such, smartphone forensics needs to consider the approaches followed by triage tools (incident response tools). This is a very challenging research area.

Table 9.2 Comparison between the smartphone forensic tools

Tools	Cellebrite	Oxygen Forensic Suite	Device Seizure	Access data	Now Secure	Final Mobile	XRY	Lantern	Secure View	Cell DEK	MOBILedit
Acquisition											
Logical	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Physical	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–
File system	✓	✓	✓	–	✓	–	✓	–	–	–	–
Password	✓	✓	✓	–	–	✓	–	✓	–	–	–
Analysis											
Malware detection	✓	✓	–	–	–	–	–	–	–	–	–
Deleted file recovery	✓	✓	✓	✓	✓	✓	✓	–	✓	✓	✓
Map view	✓	✓	✓	–	–	–	✓	✓	–	–	–
File viewer	✓	✓	✓	–	–	✓	–	✓	–	✓	–
Advanced search	✓	–	✓	✓	✓	✓	–	–	✓	–	–
Conversational view	✓	✓	✓	✓	✓	✓	–	✓	✓	–	✓
Hex viewer	✓	✓	✓	✓	✓	✓	✓	–	–	–	–
SQLite database viewer	✓	✓	–	✓	–	–	–	–	–	–	–
Plist viewer		✓	–	✓	–	–	–	✓	–	–	–
Report generator	✓	✓	✓	✓	✓	✓	✓	✓	–	✓	✓

- The fourth challenge is modification of data during investigation. According to the work proposed by Hoog, modifications taking place on a device, intentional or not, can compromise further acceptance as evidence. One of the ways that prevent this kind of changes is isolation from any network source, which, by default presents a plethora of advantages and disadvantages. There are applications where loss of connectivity triggers destruction of data.
- A research timeline has to be updated over the years, in order to preserve the ability of observing the trends within the field. This will provide researchers with quicker and more effective decision-making processes.

9.7 Conclusion

There is immense scope in the area of smartphone forensics due to the prevalence and proliferation of smartphone devices. As the use of these devices grows, more evidence and information important to investigations will be found on them. To ignore examining these devices would be negligent and result in incomplete investigations. So this will lead to true physical memory acquisitions, compared to current logical data extractions. At the same time thousands of different models exist, with different operating system and vendors, the lack of standardization in this area will lead to future work. Beyond the technical details and complexity, nontechnical issues play an equally important role. The digital forensic examiner should be constantly up to date with the field's technological developments and follow specific and established procedures.

9.8 Questions

Multiple Choice Questions

1. What is the first phase of doing smartphone forensics?

   (a) Preservation/Seizing
   (b) Acquisition
   (c) Analysis
   (d) Reports

2. If we get the device in on-state at crime location, what is the first step we should perform?

   (a) Remove SD card if it is placed in device
   (b) Isolate it from network
   (c) Switch off the device
   (d) None of the above

3. The following is not a smartphone operating system

    (a) Symbian
    (b) Android
    (c) Windows
    (d) None of above

4. TCE stands for

    (a) Trusted Computing Enterprise
    (b) Trusted Communication Environment
    (c) Trusted Computing Environment
    (d) Tested Computing Environment

5. HDFI stands for

    (a) Harmonized Digital Forensic Investigation
    (b) Horizontal Digital Forensic Information
    (c) Harmonized Defence Forensic Investigation
    (d) Harmonized Digital Forensic Interpretation

6. Smartphone forensic investigation must include following steps

    (a) Preservation
    (b) Data Acquisition
    (c) Examination and Analysis
    (d) All of the above

7. JTAG stands for

    (a) Joint Trust Action Group
    (b) Joint Test Action Group
    (c) Joint Test Active Group
    (d) Joint Test Action Graph

8. Only one of the following tools is specific to Smartphone forensics

    (a) Encase
    (b) Volatality
    (c) Oxygen
    (d) OWADE

9. UFED stands for

    (a) Universal Forensic Extraction Device
    (b) Uniform Forensic Extraction Device
    (c) Universal Forensic Examination Device
    (d) Universal Forensic Extraction Domain

10. Rooting a smartphone involves

    (a) Attain privileged access over various subsystems of the OS
    (b) Ability to alter or replace system applications or settings
    (c) Complete removal of existing version with a more recent version of OS
    (d) All of the above

Short/Long Answer Questions

1. Define smartphone forensics.
2. What are the future challenges to smartphone forensics?
3. How do I take a forensic image of an Android smartphone?
4. How can we understand the acquisition process in smartphone forensics?
5. Explain the types of acquisition.
6. Can we have the generalized/standardized framework for smartphone forensics? If yes/no, explain why?
7. Similar to rooting in Android devices, what process do we follow in the iOS? Will this process work in all latest versions of iOS devices?

References

1. Wayne J, Richard PA (2007) SP 800-101. Guidelines on cell phone forensics. National Institute of Standards & Technology, Gaithersburg, MD, United States
2. McKemmish R (2008) When is digital evidence forensically sound?. In: Advances in digital forensics IV, vol 285. Springer US, pp 3–15
3. Casey E (2011) Digital evidence and computer crime, 3rd edn. Academic Press
4. Bennett D (2012) The challenges facing computer forensics investigators in obtaining information from smartphone devices for use in criminal investigations. Inf Secur J Glob Perspect 21(3):159–168
5. Al-Zarouni M (2006) Mobile handset forensic evidence: a challenge for law enforcement. In: 4th Australian digital forensics conference, Edith Cowan University, Perth Western Australia
6. Lessard J, Kessler G (2010) Android forensics: simplifying cell phone examinations. Small Scale Digital Device Forensics (SSDDF'2010), p 12
7. ACPO (2007) Good practice guide for computer-based electronic evidence
8. Yu X, Jiang LH, Shu H, Yin Q, Liu T-M (2009) A process model for forensic analysis of Symbian smart phones. Commun Comput Inf Sci 59:86–93
9. Savoldi A, Gubian P (2009) Issues in Symbian S60 platform forensics. J Commun Comput 6(3):16
10. Breeuwsma IMF (2006) Forensic imaging of embedded systems using JTAG (boundary-scan). Digit Investig 3(1):32–42
11. Breeuwsma M, Jongh MD, Klaver C, Knijff RVD, Roeloffs M (2007) Forensic data recovery from flash memory. Small Scale Digit Device Forensics J 1(1):1–17
12. Mokhonoana PM, Olivier MS (2007) Acquisition of a Symbian smart phone's content with an on-phone forensic tool. In: Southern Africa Telecommunication Networks and applications conference (SATNAC 2007), Sugar Beach Resort, Mauritius
13. Ramabhadran A (2009) Forensic investigation process model for windows mobile devices. Available: http://www.forensicfocus.com/downloads/windows-mobile-forensic-process-model.pdf, 31 Mar 2016

14. Valjarevic A, Venter HS (2012) Harmonised digital forensic investigation process model. In: IEEE information security for South Africa (ISSA'12), Johannesburg, Gauteng, pp 1–10
15. Omeleze S, Venter HS (2013) Testing the harmonised digital forensic investigation process model-using an Android smartphone phone. In: IEEE information security for South Africa, (ISSA'13), Johannesburg, pp 1–8
16. Ayers R, Brothers S, Jansen W (2014) Guidelines on mobile device forensics. [Online]. Available: http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-101r1.pdf, 31 Mar 2016
17. Grispos G, Storer T, Glisson WB (2011) A comparison of forensic evidence recovery techniques for a windows mobile smart phone. Digit Investig 8(1):23–36
18. Quick D, Alzaabi M (2011) Forensic analysis of the android file system YAFFS2. In: 9th Australian digital forensics conference, Edith Cowan University, Perth Western Australia
19. Thing VLL, Chua TW (2012) Symbian smartphone forensics: linear bitwise data acquisition and fragmentation analysis. In: Computer applications for security, control and system engineering, vol 339. Springer, Berlin/Heidelberg, pp 62–69
20. Alghafli KA, Jones A, Martin TA (2012) Forensics data acquisition methods for mobile phones. In: IEEE 7th international conference for internet technology and secured transactions (ICITST-2012). London, pp 265–269
21. Sasidharan SK, Thomas KL (2011) BlackBerry forensics: an agent based approach for database acquisition. In: Advances in computing and communications, vol 190. Springer, Berlin/Heidelberg, pp 552–561
22. Park J, Chung H, Lee S (2012) Forensic analysis techniques for fragmented flash memory pages in smartphones. Digit Investig 9(2):109–118
23. Ahmed R, Dharaskar DRV, Thakare DVM (2013) Efficient generalized forensics framework for extraction and documentation of evidence from mobile devices. Int J Enhanc Res Manage Comput Appl, (IJERMCA'13), 2(1):7
24. Cellebrite (1999) Cellebrite UFED touch ultimate. Available: http://www.cellebrite.com/smartphone-forensics/products/standalone/ufed-touch-ultimate, 2014
25. Fedorov O (2009) Oxygen forensic suite. Available: http://www.oxygen-forensic.com/en/download/documentation, 2014
26. P Corporation (2004) Device seizure. Available: https://www.paraben.com/downloads/ds7-guide.pdf, 2015
27. A Data (n.d.) Mobile Phone Examiner Plus (MPE+)
28. Hoog A (2014) Now secure forensic suite. Available: https://www.nowsecure.com/forensics/, 2015
29. F Data (n.d.) Final mobile forensics. Available: http://www.finaldata.com/Forum2/?s=PRD&c=18&n=51, 31 Mar 2016
30. M Systemation (2014) XRY complete. Available: https://www.msab.com/xry/xry-complete, 31 Mar 2016
31. K Forensics (2014) Lantern. Available: https://katanaforensics.com/products/, 31 Mar 2016
32. S Inc (1992) Secure view. Available: http://secureview.us/kits-and-more.html, 31 Mar 2016
33. Logicube (n.d.) CellDek. Available: http://www.logicube.com/knowledge/celldek-tek#sd, 31 Mar 2016
34. C Labs (2010) Mobiledit forensic. [Online]. Available: http://www.smartphonedit.com/forensic-guide, 31 Mar 2016

Chapter 10
Cloud Forensics

Learning Objectives

- Background of cloud computing, and cloud security
- Definition of cloud forensics
- General process model of cloud forensics
- Attribution in cloud forensics
- Use of virtual machine introspection
- Research challenges in cloud forensics

Cloud computing is contemporary computing because it uses all the computing technologies in such a way that it provides everything as a service in the digital world. Cloud computing is the set of many different technologies such as grid computing, autonomic computing, utility computing, and mainly virtualization. So, assuming cloud computing as new technology is a myth. With the set of many technologies, cloud is a new business model which shows new general issues to digital forensics for both the practitioners and the community too [1]. An improvised definition of cloud computing is given by the National Institute of Standard and Technology (NIST): "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly pro-visioned and released with minimal management effort or service provider interaction" [2]. This cloud model promotes the availability and is composed of five essential characteristics, three service models, and four deployment models. The five essential characteristics are on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. The three basic service models are software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). Deployment models are private cloud, public cloud, community cloud, and hybrid cloud.

When a cloud customer uses take services from cloud service providers (CSP), there should be trust between them. Challenges for both of cloud customer and

R.C. Joshi, E.S. Pilli, *Fundamentals of Network Forensics*, Computer Communications and Networks, DOI 10.1007/978-1-4471-7299-4_10

cloud service providers are different. Cloud customer challenges include lack of transparency, high cost, and complexity of adjustment with CSP, and lack of security, auditing, and compliance. On the other hand, CSP challenges are security issues of multi-tenant users, high cost, and complexity of integration with each of cloud customers, effective disaster recovery, law enforcement issues. Beside these trust issues, Cloud Service Alliance (CSA) has produced a report of top threats in cloud computing which are enlisted as data breaches, data loss, account hijacking, insecure APIs, Denial of Service, malicious insiders, abuse of cloud services, insufficient due diligence, and shared technology issues [3]. Attack can be launched for following consequences: identity theft, fraud, extortion malware, phishing, spamming, spoofing, spyware, Trojan and viruses, Denial of service and Distributed Denial of Service attacks, breach of access, password sniffing, system infiltration, etc. There is a major fear over the loss of data (confidentiality) as their data is not directly under their control, whether the data stored does not tamper with (integrity), and what would happen if the provider was attacked (availability).

Forensics in cloud is an approach that attempts to investigate and analyze cloud security threats. This will ensure that attackers will be more cautious to avoid prosecution for illegal actions. It acts as a deterrent, reducing network crime rate and thereby increasing security. Cloud computing services are offered through network services, which are accessed over the Internet by cloud consumers, and these services are backed up by physical and virtual hardware. There are three sources from which evidence can be extracted: the client system, the network layer, and cloud service providers' management server. In public cloud, it is difficult to collect logs because of the distributed nature of the data centers. Once the data is stored, deleted, or edited in cloud environment, the logs will be generated. Without this log information, investigation cannot be progressed.

10.1 Cloud Forensic Definitions

It is stated by National Institute of Standard and Technology (NIST) that "cloud computing forensic science is the application of scientific principles, technological practices and derived and proven methods to reconstruct past cloud computing events through identification, collection, preservation, examination, interpretation and reporting of digital evidence [4]." A list of potential evidence sources for cloud forensics is also presented by Cloud Security Alliance (CSA, 2013) [5], which are given as Web server logs, application server logs, database logs, guest OS logs, host access logs, virtualization platform logs, network captures, billing records, management portal logs, DNS server logs, virtual machine monitor logs, host OS logs, and API logs.

Ruan et al. [6] have proposed a working definition for cloud forensics as the application of digital forensic science in cloud environments. Technically, it consists of a hybrid forensic approach (e.g., remote, virtual, network, live, large scale, thin client, thick client) toward the generation of digital evidence. Organizationally it involves interactions among cloud actors (i.e., cloud provider, cloud consumer, cloud broker, cloud carrier, cloud auditor) for the purpose of facilitating both internal and external investigations. Legally it often implies multi-jurisdictional and multi-tenant situations.

Barret et al. [7] have described cloud forensics with two divisions, the first one is "conducting a forensic investigation on the cloud environment," and the second one is "conducting a forensic investigation in a cloud environment." It is stated that investigation on the cloud is much than a forensic investigation in a cloud environment because of the law, court-approved methods, standard operating procedures for investigators, and the involvement of a third party.

10.2 Generic Process Model for Cloud Forensics

In this section, we propose a generic process model for cloud forensics based on the study of many existing forensic models for cloud computing. ISO 27037 standard for investigation process international standards looks for creating a common baseline for the practice of digital forensics [5].

The purpose of this standard is to assist the usability of evidence obtained in one jurisdiction by a legal process operating in another jurisdiction. ISO 27307 suggests only initial steps of forensics process: identifying, obtaining, and preserving potential digital evidence (Fig. 10.1). These processes are defined as follows:

Identification – It is a process involving the search for, recognition, and documentation of potential digital evidence.

Collection – It is a process of gathering items that contain potential digital evidence.

Acquisition – It is a process of creating a copy of data within a defined set. It is a process to maintain and safeguard the integrity and/or original condition of the potential digital evidence.

Cloud forensics ties computer forensics and network forensics together. The earlier computer forensic models focus on the investigation of a stand-alone

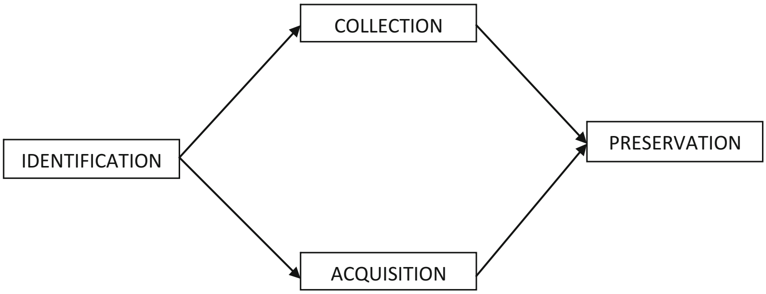


Fig. 10.1 Evidence-handling processes according to ISO 27037

Table 10.1 Computer forensics and cloud forensics

	Computer forensics	Cloud forensics
Data collection	Physical hardware	Physical hardware and virtual image
Evidence location	Client system and network system	Client system, network system, and cloud provider side
Live acquisition	Possible	Difficult
Deleted data recovery	Possible	Difficult
Cryptography	Slow	Fast
Time stamp	Accurate	Hard to keep consistency
Tools	Available	Not available
Forensic process	Developed	Under development

Table 10.2 Network forensics vs cloud forensics

	Network forensics	Cloud forensics
Logical access	Path fixed and known from source storage to target storage	Path hard to identify and may not remain fixed
Physical access	Feasible for an organization to acquire physical device and minimize impact	Digital forensic process might need to travel overseas assuming access will be granted to host server
Flexibility of methods	A variety of methods available depends on imaging requirements	Likely to be dependent on cloud providers, except for IaaS
Segregation of evidence	Normal practice within single organization	Physical acquisition not likely an option; cloud providers assistance for logical acquisition
Legal implications	Simple ownership; relatively straightforward	Ownership; jurisdiction; multi-tenancy for physical acquisition

computer and interpretation of data stored in it. Computer forensic investigator has the advantage of specialized tools which the attacker lacked whereas the cloud investigator and the attacker are at the same skill level. It is required to develop a process model specific to cloud forensic investigation totally at a different plane concerning computer forensics. Comparison of computer forensics and cloud forensics is shown in Table 10.1.

Network forensics is the capture, storage, and analysis of network events. A comparison of network forensics and cloud forensics is shown in Table 10.2.

The proposed model is generic as it aggregates many of the phases available in the digital forensic models but builds on those phases which are specific to cloud forensics. The framework is shown in Fig. 10.2. We give a detailed explanation for each of the following phases:

(i) Verification

It is a legal requirement to do investigation freely. Service-level agreement should not be broken. The investigator needs full authorization to perform forensics at every

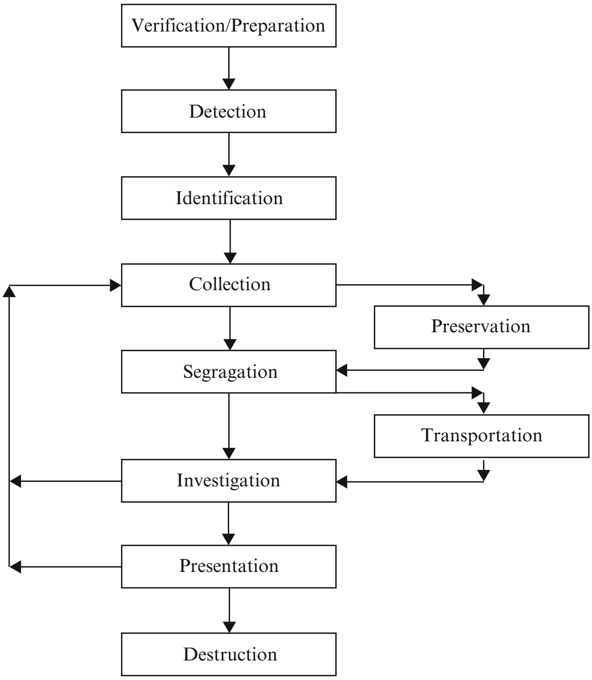


Fig. 10.2 Generic process model of cloud forensics

level. It needs to understand the "what, where, when, who, why, and how" of an investigation and to determine the boundaries of an investigation.

(ii) Detection

Where is the data and what data is available. If any unauthorized event occurs, it should be analyzed at a time. We can set up tools or alert generator on sensitive places. When any anomalous activity is generated, data can be collected at that time. Virtual machine introspection also helps to collect data from virtual machines, hypervisor, and memory mapping.

(iii) Identification

To identify the place, where the data is present, and what data is present, who is the victim, and who has broken SLA, all these questions should be answered in this section. Martini et al. [1] have also discussed this phase in detail adding with preservation. But with identification process, it is not possible at all to preserve the evidence simultaneously.

(iv) Collection and Preservation

A collection of the data is an important part mainly in cloud system because of the volatile and non-volatile nature of the data. While collecting the data, it is needed to have an original copy of it for future reference. This collected evidence is very much sensitive, so it is needed to preserve it simultaneously. Preservation is necessary because the collected evidence may be altered or deleted. Most of the authors have included this phase in their model.

(v) Segregation and Transportation

Segregation of the collected evidence is necessary from the time-saving perspective in examination and analysis phase. All the collected data are not necessarily useful. So, segregation according to the format of data, files, and folders is necessary. This phase helps in examination and analysis step. By needs, available data can be transported to other storage media or labs.

(vi) Investigation

After reducing the redundancy and with common file format in segregation phase, the data is now examined and analyzed. This is more analytical part in forensics. In this phase, by collected evidence, different cases are studied. This phase is the main part, and it is added all discussed phases.

(vii) Presentation

After examination and analysis part, a report is generated which includes the conclusion of the whole process. It also includes presenting the evidence to the court of law. It is unchanged and cannot be ignored anyway. Finally, one document is prepared for future reference so that if there is any case that arises again in a similar way, this document can help.

(viii) Destruction

After completing the forensic process, it holds a significant quantity of sensitive data, and, if this information is mistaken, it can be exploited by the malicious users. Data destruction is the safe cleanup or physical devastation of the digital data so that the data is no longer recoverable.

10.3 Investigation of Cloud Infrastructures

The cloud system is represented as the grouping of three types of system components for delivering cloud services, i.e., physical resource layer, resource abstraction layer, and service layer, as shown in Fig. 10.3. This section presents the survey of investigation of cloud infrastructure. Physical resource layer includes hardware computing resources such as computers (CPU and memory), networks (routers, firewalls, switches, network links, and interfaces), and storage components (hard disks). Resource abstraction and control layer contain the system components that cloud providers use to provide and manage access to the physical computing resources through software abstraction.

Resource abstraction components typically include software elements such as hypervisors, virtual machines, virtual data storage, and other computing resource abstractions. The service layer is where cloud providers define interfaces for cloud

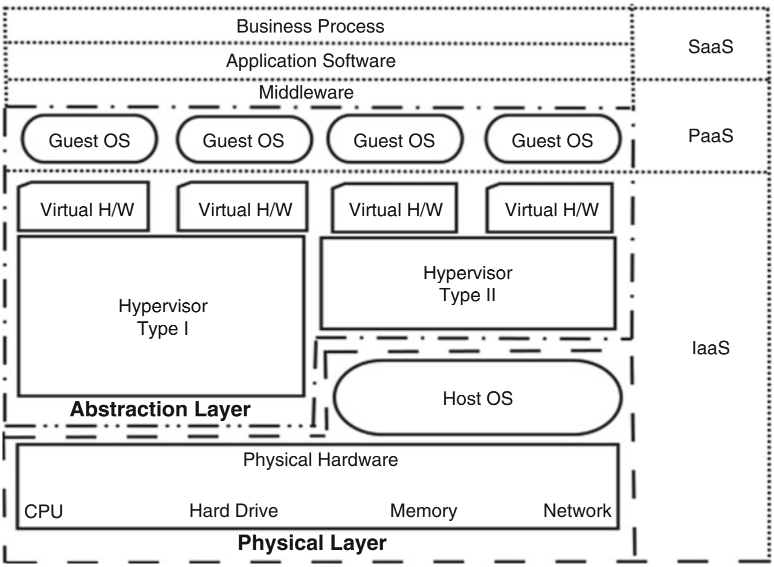


Fig. 10.3 Cloud computing environment

consumers to access the computing services. Access interface of each of the three service models is provided in this layer. It is possible, though not necessary, that SaaS (application layer) applications can be built on top of PaaS (middleware layer – libraries, database, and Java virtual machine) components, and PaaS components can be built on top of IaaS (OS layer) components.

Taylor et al. [8] have presented a model of a forensic investigation involving multiple layers in a cloud environment such as hypervisor level, cloud platform level, cloud application level, and cloud client layer. Under the linear process, authors have suggested four phases of computer forensic investigation which include identifying potential sources of evidence, maintaining integrity at the time of evidence extraction, analysis on a live system (i.e., incident response), and forensic laboratory using forensic tools like EnCase[1], Forensic Tool Kit[2], etc. Challenges at acquisition phase are discussed such as remote Web access, virtualized platform, and third-party location physical seizure and identifying the suspicious action in a cloud environment. It is suggested that computer forensic tools like EnCase and FTK in cloud platform cannot perform all functions of investigation due to the newly written application. The purpose of the forensic investigation such as finding a reason for unauthorized access or suspected money laundering should be clearly mentioned. Authors suggested that if auditing is maintained by cloud system, the

[1] http://www.guidancesoftware.com/encase-forensic

[2] http://www.accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk

forensic investigation may be easier for the analyst. Installing forensic agent in both of the cloud deployment model (public and private) has recommended for a better result in investigation process.

10.4 Cloud Forensic Attribution

When the Internet is aimed to be attacked, its difficulty makes sense of who did it. The truth of worldwide hostility on the Internet will change how we will approach barrier in a better way. Attribution is the procedure of deciding and finding the character and cause of digital attack [9]. Various literatures are available to perform attribution of computer attacks and network attacks. But, until now there is no exhaustive detail available about cloud attack attribution. As we have discussed above that there is a need for specific tools and techniques to resolve cloud forensics issues, we need to develop a new methodology for attribution in cloud forensics.

Attribution is not only limited to know the source of the attacker or to identify the attackers, but this process is started with the data extraction, metadata analysis, and event reconstruction. Attribution in a cloud environment is more difficult because of the complex architecture of cloud computing, data center locations, multi-tenancy, CSP dependency, hypervisor environment, and lack of training and support. In this section, a survey of attribution techniques is presented.

Wheeler et al. [10] have presented definition, techniques, and issues in the attribution of computer attacks. The definition of attribution is to determine the identity or character, place, or position of an attacker or an attacker's intermediary. Computer attack or network attack attribution techniques may not be applicable in cloud computing environment, but we are evaluating some of these techniques, and we search few areas in cloud computing, where we can apply these techniques.

Cohen et al. [11] have surveyed attack attribution techniques at three levels. At first, traceback mechanism is discussed which is explained with link testing, packet marking methods, source path identification, and remote monitors. The second level of attribution method includes reflection traceback, internal monitoring, logging, snapshot, network traffic, zombie traceback, and physical traceback. The third level of attribution is analysis methods such as document examination, e-mail, and chat investigation, and the use of Bayesian networks, hidden Markov models, and support vector machines, etc.

Hunker et al. [12] have discussed attribution challenges such as Internet design, regulatory concerns, a technical obstruction such as virtual networks, log destruction, users' anonymity, and sophisticated attacks. In cloud computing environment, these challenges are more serious as cloud architecture is more complex due to abstraction layer and data center networking.

Dacier et al. [13] have introduced a new method for attack attribution that is applied to honeynet data set. Various steps are explained to perform attribution such as event identification, event features, feature selection; graph-based clustering, and aggregation.

Fu et al. [14] have proposed pay-as-you-go traceback model and discuss how an investigation can be performed. The application of digital forensics to cloud computing is explored. Traceback through the cloud and analyzing optimization of cloud computing investment for maximum traceback are examined. Results using Amazon EC2 over Tor for which different algorithms were used. The evaluation of network forensic strategy through cloud computing is also shown.

Knake et al. [15] have discussed two types of attribution problem in the cyber world, which are Internet based and non-Internet based. The responsibility of government, rules, and regulations are also explained. Clark et al. [16] have presented a general discussion of attribution on the Internet. Authors have discussed a brief history of the Internet, class of attacks such as botnet-based attacks, and multistage attacks; finally different aspects of attribution are explained which are the type, time, investigator, and jurisdiction. Two levels of attribution are discussed as machine level and application level. Idziorek et al. [17] have proposed an attribution methodology to identify malicious clients participating in a Fraudulent Resource Consumption (FRC) attack. The anomaly detection methodology is explained to perform attribution, which can accurately identify malicious clients – based on IP addresses from that of legitimate clients.

Shakarian et al. [18] have proposed a new framework called as Intelligent Cyber Attribution (InCA) for attack attribution in cyber world which uses a combination of a various algorithms such as reasoning model, logistics, and a probabilistic model.

Giffin et al. [19] have proposed a model for attribution in virtualized environment to detect malicious activity. Incremental innovation in attribution methods of malicious activity is also discussed such post-attack investigation, code analysis, information flow analysis, virtual machine introspection, and data recovery. Table 10.3 shows different attribution techniques from computer attack and network attack attribution. These techniques [10] can be applied in some of the areas of cloud forensics.

10.5 Investigation Using VMI in Cloud Environment

Attacks focusing on hypervisors are serious since they may crash the hypervisors including all guest OS or virtual machines. A vulnerable hypervisor can permit the attacker to make each guest machine installed on a virtual machine monitor (VMM) to be vulnerable. One conceivable result is an increment in the asset utilization of a VM that causes a Denial of Service attack over service provider's server, this means when more virtual servers are included, more issues will be exacerbated.

Virtual machine introspection (VMI) is used for investigating real-time events of the virtual machine and assures whether the virtual system is running properly. Virtual machines take into account such investigation without interruption of the monitored VM. At first, VMI was proposed in [20], which defines VMI as "examining a virtual machine from the outside with the end goal of dissecting

Table 10.3 Attribution techniques in cloud forensics

Techniques	Description	Use in cloud forensics
Store logs	Logs are stored of all activities	In CF, it can help to extract valuable information and at the time of event reconstruction
Reconfiguration and observation	In a small network, these techniques can work to change overall system	Due to the complex architecture of cloud computing reconfiguration is not possible. However, monitoring can be performed which can later help in metadata analysis
Insert host monitor functions	It is performed using hack back without permission of the owner and that need legal control	An agent can be installed in each of the cloud systems if legal actions are verified
Match streams	This is stream matching of the packets or data to recognize between actual data and modified data	However, these techniques can be implemented using cryptography, and it helps preservation phase too
Intrusion detection system	Placement of IDS in existing system appropriately may help to investigation	IDS implementation in cloud environment will be difficult, and data collection for attribution purposes will be more difficult due to the third-party location and dependency of CSP's
Filtering approach	This technique is used in network to pass authenticated data, and it helps attribution to avoid additional storage space	This approach can help in same way as in network, but sometimes loss of other data may affect the history of events
Snapshots	Snapshot of running computer system, live memory can help to attribution	VM snapshot, live machine snapshot, and live RAM capture are possible in a cloud environment

the software running on it." VMI collects low-level information that is acquired externally. This low-level information collects virtual address, system call table, and Interrupt Descriptor Table (IDT).

VMI can be implemented in a cloud environment for the purpose of the investigation because every cloud service provider uses hypervisor. A sample architecture is shown in Fig. 10.4, where client access services are offered through the Internet. Each of the compute nodes is monitored by virtual machine introspection, which provides memory information, hardware events, and virtual CPU register information. A popular VMI tool is established with administrative privileges. VMI helps us to extract useful data from an untrusted guest machine and can provide information about the virtual disk, memory size, network connection, and hardware events. Memory has physical address information, virtual address information, and application symbols. Register's information can also help to monitor virtual machine. When we access memory pages, we should know what parameters are being monitored by virtual machine and what processes are running inside a virtual machine. The user application can also be observed that what request or functions

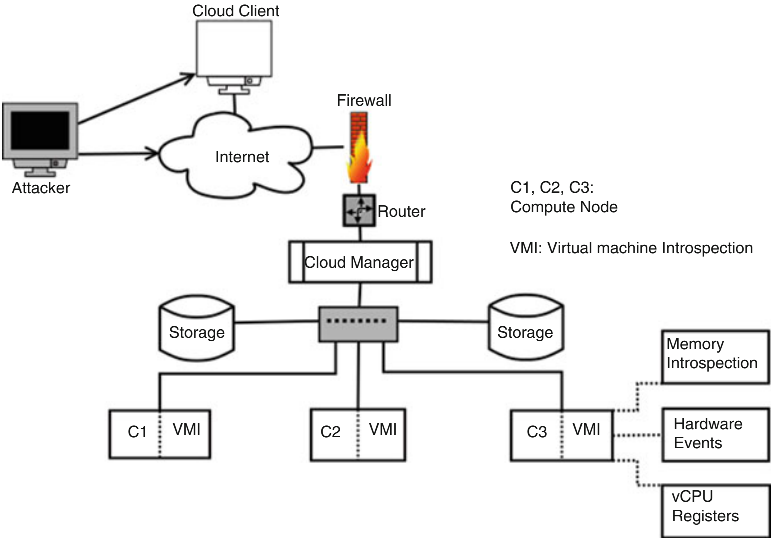


Fig. 10.4 Use of VMI in cloud environment

are being used. When memory introspection is performed for the static virtual machine, there are lots of facilities where we can introspect with flexible timing slot, filtration process, and use of external analysis tools. However, for the live virtual machine, we have limited resources and less timing to introspect.

10.6 Cloud Forensic Challenges

Cloud forensic challenges are described in this section. NIST has categorized cloud forensic challenge in nine parts which are again subdivided in 65 research challenges. Here, we are describing these challenges to make a better understanding of cloud forensic gaps.

(i) Architectural Issues (Multi-Tenancy, Data Segregation, Provenance):

Architectural issues of cloud forensics include single point of failure, detecting malicious activity, real-time investigation, lack of transparency, multi-tenancy, data segregation, and secure provenance.

(ii) Evidence Collection (Evidence Location, Identification, and Segregation, Live Data Recovery):

To locate the location of digital evidence and to identify the evidence and segregation are the major challenges in evidence collection. These challenges are subdivided as data controlling, a chain of dependencies, imaging and isolating data, resource abstraction, and finding application details [4].

(iii) Evidence Analysis (Evidence Correlation, Metadata Analysis):

Evidence analysis includes evidence correlation, event reconstruction, time stamp synchronization, log format unification. Evidence analysis is to be done through machine learning techniques, soft computing, statistics, and data mining approaches.

(iv) Anti-Forensics:

Anti-forensics is a set of actions which goal is to prevent proper forensic investigation process or make it more complex. It intended to reduce the quantity and quality of digital evidence. Compromising events, information disruption, and misleading investigation process are the main goals of anti-forensics. Anti-forensics is subdivided into following parts: data hiding, obfuscation and encryption, data forgery, data deletion and physical destruction, analysis prevention, etc.

(v) Incident Responses:

Finding out what happened, how it happened, and who did it, are the main problem in cloud forensics because cloud computing has distributed environment, and it is very critical to determine solutions of these problems at the instant time. To maintain this, the investigator should not be confused to take a decision about what to do in quick responses.

(vi) Service-Level Agreement (SLA):

An SLA represents the understanding between the cloud consumer and cloud provider about the expected level of service to be delivered and, if the provider fails to deliver the service at the level specified and the compensation available to the cloud consumer [21]. Considering the distribution of control between CSP and customer, it becomes apparent that it remains almost impossible for the customer to verify the actual performance of these agreements [22]. Due to the lack of customer awareness, there are limited rules and regulations regarding forensic investigations. Most cloud customers are unknown of these issues that may arise in a cloud computing [23].

(vii) Lack of Standards:

While collecting evidence in cloud forensics, rules and acts must be followed strictly. In most of the cases, privacy is not preserved, while the investigation is performed. So forensic investigation can only be performed under the act which is already defined by different established standard organizations. There exists no standardized logging format for the cloud. Each provider logs in different formats, making log crunching for forensics difficult in the case of the cloud. It is needed to develop for these policies for which records can be kept in the standardized format [24].

(viii) Lack of Tools and Framework:

There is a lack of tested and certified tools which can be able to perform cloud forensics. Until now there is no tool developed completely by which cloud forensics can be performed very well. So it is needed to develop such tool which provides a complete package to install and run on a system and can monitor the process of investigation.

One general framework is needed to perform all phases of cloud forensics. There is no standard framework developed which can be used. A lot of work is still needed to build an essentially supported structure for cloud forensics which can be implemented practically [24].

10.7 Conclusion

Different views of many authors, about the definition of Cloud forensics, were presented. Cloud forensics is the extended version of network forensics which includes various dimensions of digital forensics. A general process model for Cloud forensics has been given explaining the proactive and reactive approaches of investigation. Forensic investigation at various cloud service layers (physical layer, resource abstraction layer, and service layer) is also described for better understanding. We have introduced Cloud forensic attribution and surveyed the previous work of cyber-attack attribution. Previous attribution techniques are described to support forensics in cloud computing environment. A model has been explained which uses virtual machine introspection (VMI) technique in cloud environment. VMI tool can provide low level information at hypervisor level. This mechanism can be useful in a few cases when we need information such as memory information, hardware events, and virtual CPU register. At last Cloud forensics challenges are explained which are broadly classified as nine categories by NIST. This provides a starting point for understanding security and forensics in Cloud Environment.

10.8 Questions

Multiple Choice Questions

Select the most suitable answer for the following questions:

1. CSA stands for ____.

   (a) Cloud Security Alliance
   (b) Cloud Security Attribution
   (c) Cloud Service Alliance
   (d) Cloud Sensor Application

2. A cloud customer uses take services from ____.
   - (a) Cloud Service Alliance
   - (b) Software Developers
   - (c) Software-Defined Networks
   - (d) Cloud Service Providers
3. International Standards ISO 27037 looks for ____.
   - (a) Creating a common baseline for the practice of digital forensics
   - (b) Attribution
   - (c) Attack simulation
   - (d) Anti-forensics
4. Attribution is the procedure of ____.
   - (a) Deciding and finding the character and cause of digital attack
   - (b) Acquisition of evidence
   - (c) Service-level agreement
   - (d) Software development
5. VMI collects ____.
   - (a) High-level information
   - (b) Software programs
   - (c) Only logs files
   - (d) Low-level information
6. Anti-forensic prevents ____.
   - (a) Forensic investigation process
   - (b) Attacker to attack
   - (c) Security threats
   - (d) Malicious attacks
7. SLA stands for ____.
   - (a) Service-level agreement
   - (b) Security-level alliance
   - (c) Software-level awareness
   - (d) Software-level agreement
8. The cloud service models are ____.
   - (a) SaaS, PaaS, and IaaS
   - (b) Public and private
   - (c) Able to perform on-demand services
   - (d) Cost-effective

9. Attribution in a cloud environment is more difficult because ____.
   (a) Service-level agreement
   (b) Complex architecture of cloud computing
   (c) Cloud actors
   (d) Cloud consumer
10. Multi-jurisdictional and multi-tenant situations are ___.
   (a) Analytical issues
   (b) Functional issues
   (c) Architectural issues
   (d) Legal issues

Short-Answer Questions

Write the brief answers of following questions:

1. What are the security threats to cloud computing?
2. What are the major issues in cloud computing environment? Define each of them.
3. Define cloud forensics. How is it different from cloud security?
4. Explain the potential evidence sources during cloud forensic investigation.

Long-Answer Questions

Write in detail answer of following questions:

1. Write the general process model of cloud forensic. How they are interrelated to each other?
2. What are the major research challenges in cloud forensics? Explain each of them.
3. Explain the use of virtual machine introspection in cloud forensics.

References

1. Martini B, Choo KKR (2012) An integrated conceptual digital forensic framework for cloud computing. Digit Investig 9(2):71–80
2. Badger L, Grance T, Patt-Corner R, Voas J (2011) Draft cloud computing synopsis and recommendations
3. CSA (2013) The notorious nine: cloud computing top threats in 2013
4. NIST (2014) NIST cloud computing forensic science challenges, NIST draft NISTIR 8006
5. CSA (2013) Mapping the forensic standard ISO/IEC 27037 to Cloud Computing
6. Ruan K, Carthy J, Kechadi T, Baggili I (2013) Cloud forensics definitions and critical criteria for cloud forensic capability: an overview of survey results. Digit Investig 10(1):34–43
7. Barrett D, Kipper G (2010) Cloud computing and the forensic challenges. In: Virtualization and forensics, 1st edn. Elsevier, Boston, ch. 10, sec. 3, pp 197–209
8. Taylor M, Haggerty J, Gresty D, Lamb D (2011) Forensic investigation of cloud computing systems. Netw Secur 2011(3):4–10
9. Computation-Institute (2007) Forensics, traceback, and attribution. [Online]. Available: https://wiki.ci.uchicago.edu/FranksProjects/ForensicsTracebacksAttribution, 19 Sept 2014

10. Wheeler DA, Larsen GN (2003) Techniques for cyber attack attribution. DTIC Document
11. Cohen D, Narayanaswamy K (2004) Survey/analysis of Levels I, II, and III attack attribution techniques, vol 27. Cs3 Inc
12. Hunker J, Hutchinson B, Margulies J (2008) Role and challenges for sufficient cyber-attack attribution. Institute for Information Infrastructure Protection
13. Dacier M, Pham VH, Thonnard O (2009) The WOMBAT attack attribution method: some results. In: 5th international conference on information systems security (ICISS), Kolkata, India, pp 19–37
14. Xinwen F, Zhen L, Wei Y, Junzhou L (2010) Cyber crime scene investigations through cloud computing. In: IEEE 30th international conference on distributed computing systems workshops (ICDCSW), pp 26–31
15. Knake RK (2010) Untangling attribution: moving to accountability in cyberspace, T. C. o. F. Relations, ed., 2010
16. Clark DD, Landau S (2010) Untangling attribution. Harv Nat'l Sec J 2:323
17. Idziorek J, Tannian M, Jacobson D (2012) Attribution of fraudulent resource consumption in the cloud. In: IEEE 5th international conference on cloud computing (CLOUD), pp 99–106
18. Shakarian P, Simari GI, Moores G, Parsons S, Falappa MA (2015) An argumentation-based framework to address the attribution problem in cyber-warfare
19. Giffin J, Srivastava A (2010) Attribution of malicious behavior. In: Information systems security, vol 6503. Springer, Berlin/Heidelberg, pp 28–47
20. Garfinkel T, Rosenblum M (2003) A virtual machine introspection based architecture for intrusion detection. In: Network and distributed systems security symposium, pp 191–206
21. Jansen W, Grance T, Guidelines on security and privacy in public cloud computing. NIST Special Publication, pp 800–144
22. Birk D, Wegener C (2011) Technical issues of forensic investigations in cloud computing environments. In: IEEE sixth international workshop on systematic approaches to Digital Forensic Engineering (SADFE), 2011, Bochum, Germany, pp 1–10
23. Ruan K, Carthy J, Kechadi T, Crosbie M (2011) Cloud forensics. In: 7th IFIP WG 11.9 international conference on digital forensics, Orlando, FL, USA, pp 35–46
24. Mishra AK, Priya M, Emmanuel SP, Joshi RC (2012) Cloud forensics: state-of-the-art and research challenges. In: International symposium on cloud and services computing, pp 164–170

Index

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