The Seybold Report

ISSN 1533-9211

DOI 10.17605/OSF.IO/XC23Y

THE EFFICIENT PROACTIVE APPROACH FOR NETWORK FORENSICS THROUGH CRYPTOGRAPHIC AND DATA MINING TECHNIQUES

RASHMI NILESH MALVANKAR

Research Scholar, Sir Padampat Singhania University, Udaipur, Rajasthan.
Email: rashmi.malvankar@spsu.ac.in

Dr. ANAND KUMAR BHASKAR

Assistant Professor, Sir Padampat Singhania University, Udaipur, Rajasthan. Email: anand.bhaskar@spsu.ac.in

Dr. AMIT JAIN

Professor, O P Jindal University, Raigarh, Chhattisgarh. Email: amit.jain@opju.ac.in

Dr. ARUN KUMAR

Professor, Sir Padampat Singhania University, Udaipur, Rajasthan. Email: arun.kumar@spsu.ac.in

Abstract

With the advancement and popularity of computer networks, a growing number of devices are being added to global internet connectivity. Furthermore, more advanced tools, processes, and strategies are being applied to improve the worldwide internet connection. It is also worth noting that individuals, businesses, and corporate organizations are fast realizing the importance of computer networking. However, the popularity of computer and mobile networking has several problems, the most of which are related to security and data breaches. Every day, cybercriminals investigate and create complex methods of entering and compromising the security of individual and corporate networks. This implies that cyber or network forensic investigators must be equipped with the appropriate tools for detecting the type of security vulnerabilities as well as the capacity to accurately identify and capture cyber-related criminals. As a result, the major goal of this research is to give a full analysis of the notion of network forensic investigation as well as to describe the methodology and instruments used in network forensic investigations. Finally, in a network forensics inquiry, this research presents an evaluative analysis of the relevant literature review.

Keywords: Network Forensic, Proactive, Cybercrime, Encryption, Data Mining, Network Traffic

1. INTRODUCTION

The growth of computer networks and the internet has offered several chances for the commission of cybercrime. Throughout the world, numerous computing devices are linked to a complex mesh of computer networks. Cyber attackers are always developing complex techniques to commit cybercrime. The victims pay a high price because of the nature and type of crime [1]. In rare cases, cybercrime not only results in large financial losses but may even render the victimized firm unusable. As a result, it is critical to have a process in place to conduct the appropriate investigation and audit in order to determine the course and perpetrators of the connected cybercrimes. The process is known as network forensics in the context of cyber-criminal investigations.

ISSN 1533-9211

DOI 10.17605/OSF.IO/XC23Y

Network forensics is a digital forensic procedure that comprises the investigation, analysis, and surveillance of computer networks to find vital information that aids in the capture of cybercriminals [2]. Network forensics also aids in the collection of essential and legal information, evidence, and intrusion detection trails. In essence, network forensics assists a cyber-forensic investigator in monitoring network traffic and identifying harmful material inside network traffic. Because network forensics is data-centric, it is not limited to the analysis of network traffic. Instead, it is connected with similar ideas such as mobile forensics, memory forensics, and host-based forensics [1].

There are many network security tools, such as firewalls and intrusion detection systems, that are used to handle network attacks, but they have many limitations, such as the inability to protect the system against attacks that bypass them, the inability to protect the system against internal threats, and the inability to handle new attacks. The preventive system investigates the source of the assault and prosecutes the skilled assailants. Network forensics [30] provides such a preventive technique. Network forensics isn't just another term for network security; it's an all-encompassing phase of network security in which data for legal investigations is acquired from numerous security instruments. A trigger is something that begins with A trigger is an occurrence or incident that alerts the organization to potentially harmful behaviors by known or unknown individuals. This might be either reactive or proactive. There are two types of network forensics methodologies: proactive and reactive. Proactive network forensics is a live investigation that controls network forensic phases throughout a stabbing.

2. RELATED WORK

This section will examine related literature on collecting live network traffic data and generating pre-processed data sets from raw network traffic. Also covered will be the numerous sorts of data mining methods that may be used on datasets.

The major purpose of Praveen P. Naik and Prashantha S. J's method was to identify infiltration using data mining techniques.

The input data set was in KDD-CUP format. After that, the dataset was separated into two parts: training data and testing data. Then, using the training data, the K-means clustering method was applied to k subsets, where k is the number of clusters necessary for grouping. Following the development of K-cluster neuro-fuzzy (FNN), each k-cluster was supplied with input. The neuro-fuzzy output was fed into the support vector machine (SVM) classifier. Finally, following classification, i.e., using SVM, it was possible to identify whether there was intrusion in the supplied data set. The fundamental purpose of this strategy, according to Amine Boukhtouta and Nour-EddineLakhdari [3], was to identify malicious traffic at the network level. In this method, they first gathered harmful data using a dynamic malware analysis tool and then recorded the raw network traffic as pcap files. They then gathered non-malicious traffic from a DARPA [8] dataset and classified it as "regular." Both infected and regular pcap files were merged and subjected to feature extraction. Later, multiple machine learning methods were used to create various classifiers capable of detecting malicious traffic at the network level. By collecting encrypted traffic, our strategy outperformed other approaches

ISSN 1533-9211

DOI 10.17605/OSF.IO/XC23Y

significantly. According to David Mudzingwa and Rajeev Agrawal [4,] the rise in security breaches in computer systems and computer networks has led to an increase in the number of security tools that investigate ways to guard against these breaches. Intrusion detection and prevention systems are among these tools (IDPS). This work aims to provide a realistic method for evaluating both hardware and software based IDPS using the freely accessible open source tools Tomahawk and Wireshark. Mrs. Ghatge Dipali D's [5] approach's major purpose was to identify network intrusion using various data mining methods such as Decision tree and K-means algorithms. She made use of the DAPRA data set, which was used for both training and testing. Following that, the DAPRA dataset was pre-processed in order to extract meaningful information from raw network data. Following pre-processing, K-means and decision tree algorithms were used to pre-processed data to detect anomalous and typical traffic. T. Subbhulakshmi1, S. G. Keerthiga2, and R. Dharini3 [6] developed the Intelligent Multi Layered Attack Classification System (IMLACS), which assisted in detecting and categorising intrusions with high accuracy. The suggested technique captures packets as they go over the network and extracts useful information from those packets. Following that, appropriate characteristics were recorded in a file. This file was fed into a binary classifier called support vector machine (SVM). The output of SVM filters the records that are detected as an attack, and this is fed into neural networks for training and testing. The output of the neural networks was fed into the fuzzy inference system (FIS). It recognised the sort of assault based on the rules observed in FIS. In this approach, a real-time dataset was utilised as an input for multiple categorize systems.

The major purpose of this strategy, according to S. Prayla Shyry [7], was to identify bots in the network using K-means clustering algorithms. Bots are computer systems or servers that launch different sorts of assaults, such as denial of service attacks, spam email attacks, guess password attacks, and so on. The botminer algorithm was used in this manner. To begin, network traffic was gathered with network capturing tools. Following collection, essential information such as source IP, destination IP, source port, destination port, protocols, and so on was retrieved from the acquired data. Only packets with the syn (synchronization) flag enabled and packets that were acknowledged were filtered, and data such as flow per hour, bytes per hour, packets per hour, and so on were computed. Following that, the mean and variance were computed, and the K-means clustering technique was used. Finally, it filtered attacked and regular packets after clustering.

3. DATA COLLECTION

We may capture various network data using network capturing programs such as wireshark, tshark, and tcpdump. Until now, wireshark has been used for research since it is platform free. Wireshark is a network protocol/packet analyzer. It is platform agnostic, which means it can run on any operating system, including Windows, UNIX, and Linux. It will attempt to collect network packets and display as much detail as possible about the packet data. It is open source and free to use. It is used for network maintenance, investigation, software and communications protocol creation, and teaching. It is capable of capturing both network and wireless traffic. It captures packets using the pcap library. Packets can also be exported as XML (PDML),

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/XC23Y

PostScript, CSV, or plain text files. Pcap captures whole packets that include some particular information. Some fields are derived from this information in order to examine each individual packet. Data is pre-processed in order to extract only important properties. Are essential throughout the data mining process. Data that we collect when we collect live network traffic, we get raw audit data. Then it must be converted to KDD format so that it may be utilized as input for numerous data mining algorithms. Java APIs for handling.pcap files from wireshark recorded files:

1. JPCAP:

- Developers can create their own packet capture programmes using the jpcap API.
- Downloads live, unprocessed packets from the network.
- Captured packets are read from those offline files after being saved to an offline file.
- Before being sent to the application, packets are filtered based on criteria that define the users.
- Sending raw packets to the network. Java and C can be used to implement it.

2. JNETPCAP:

- It is a Java wrapper for the native libraries of libpcap and WinPcap.
- It can read.pcap files.
- This API supports a number of classes, including JFlowKey, JFlow, Pcap, LOOP INFINITE, etc.

Wireshark capture (.pcap) files are processed in this study using the JNETPCAP API. The guidelines for categorizing various attacks are as follows:

Rule	Label
If(protocol==ICMP)&&(Echo_Request_count>Threshold)	Ping Attack
If(protocol==TCP)&&(Reset_count>=1)&&(No_Packets>Threshold)&&(Invalid_pwd_count>0)	Guess Password
If(protocol==TCP)&&(syn==low)&&(port==unequal) &&(IP==unequal)&&(target==TCP)	TCP flood
If(protocol==TCP)&&(syn==high)&&(port== unequal) &&(IP==unequal)&&(target==TCP)	SYNflood
If(protocol==TCP)&&(syn==low)&&(port== unequal) &&(IP==equal)&&(target==TCP)	Backattack
If(protocol==TCP)&&(syn==high)&&(port==unequal) &&(IP==equal)&&(target==TCP)	Backattack
If(protocol==TCP)&&(syn==low)&&(port==equal) &&(IP==equal)&&(target==TCP)	Landattack

The process of grouping a collection of objects into classes of related objects is called clustering. There will be one group, which will be made up of a collection of related items. The main benefit of clustering is that it isolates pertinent qualities that describe groups of various kinds and is adaptable to any changes. In this study, k-means was used for clustering. Similar attacks tended to cluster together. The converted data set that we acquired after applying the previous step, i.e., step B, was subjected to preprocessing before being subjected to the k-means

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/XC23Y

algorithm. The fundamental goal is to group attack traffic and regular traffic together. Depending on their characteristics, attacks of various kinds were grouped together.

4. DATA INTEGRITY

To preserve the integrity of the captured data through the network using cryptographic techniques.

1. Collect the data from its previous phase.
2. Encrypt the data using the Elliptic curve cryptography algorithm.
3. Compress the encrypted file using Canonical Huffman Encoding
4. Preserve compressed data.
5. Decompress the received file using Canonical Huffman Encoding
6. Decrypt the data using Elliptic curve cryptography algorithm and used for further examination

Encryption

Elliptic Curve Cryptosystem

Elliptic curves are generated using discrete logarithms over a finite field. In a highly sophisticated data system, public-key algorithms generate a method for distributing keys among a large number of participants or entities. The majority of public-key cryptosystems are built on arithmetic in finite fields, which are algebraic structures featuring addition and multiplication operations with inverses. The error correcting code constructs a finite field from the set of solutions to the elliptic curve equation and the additive identity that corresponds to the goal at infinity. ECC is effective because it is thought to be more durable to encrypt, for example, distinct logs over finite fields of code rather than inside the underlying whole number finite fields.This indicates that key sizes in code will be less than key sizes in cryptosystems supported by various fields. However, ECC is not known to be more durable than the other method.

Algorithm:

Step1:

Consider a message ‘Pm’ sent from A to B.

‘A’ chooses a random positive integer ‘K’, a private key ‘nA’ and generates the public key PA=nA*G and produces the cipher text ‘Cm’ consisting of pair of points Cm= {kG,Pm+kPB}

Where G is the base point selected on the Elliptic Curve, PB=nB*G is the public key of B with private key ‘nB’

Step2:

To decrypt the ciphertext, B multiplies 1st point in the pair by B’s secret & Subtract the result from the 2nd point Pm + kPB - nB(kG) = Pm + k(nB G) – nB(kG)=Pm

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/XC23Y

Compression

The problem with encrypting a text file is that the encrypted file is much larger than the original text file; therefore, we compress it. Text compression is a cryptography technique that compresses text so that it takes up less storage space. To meet all of these security aims, the standard ECC encryption technique requires time. As a result, a compression method is required to speed up the process. Compression is a cryptographic paradigm that decreases the amount of space necessary to keep data while also reducing the time required to transmit it. The primary goal of data compression is to represent a source in as few bits as feasible while ensuring that the minimal need for reconstruction of the original is met.

Canonical Huffman Coding

The bit array module is used to represent binary data and provides a very fast method of writing to file. The output file is opened in wb, binary-mode' for writing, and a string of binary values is supplied into a bit array. Bitarray has a helpful method for encoding and decoding data according to a codebook utilizing prefix rules, which makes it ideal for Huffman coding. -- firstWhen converting data from a file, BitArray offers a 'encode' function that, given a dictionary of characters and their corresponding encodings, will compress the data, which can then be put into a file. Because writing to a file is in bytes, an 8-bit representation, 0's are added to the string to guarantee the amount of bits is a multiple of 8. This allows the decoder to recognize when the codebook component of the compressed data begins. The bitarray has a function that estimates the amount of unused bits. This number of 0s is then appended to the beginning of the output, along with a numerical value indicating how many there are in front of it. This number may be read at the start of decompression and the right number of 0s deleted to guarantee the codebook and subsequent data are read appropriately.

Canonical Huffman Encoding Algorithm

Encoding:

To assign a canonical Huffman code to a set of symbols, supposing that symbol i is to be assigned a code of l_i bits, that no codeword is longer than *maxlength*, and that there are n distinct symbols,

1. For $l \leftarrow 1$ to *maxlength* do
   Set $numl[l] \leftarrow 0$.
   For $i \leftarrow 1$ to n do
   Set $numl[l_i] \leftarrow numl[l_i] + 1$.
   Number of codes of length l is stored in $numl[l]$.
2. Set $firstcode[maxlength] \leftarrow 0$.
   For $l \leftarrow maxlength - 1$ downto 1 do
   Set $firstcode[l] \leftarrow (firstcode[l + 1] + numl[l + 1])/2$.
   Integer for first code of length l is stored in $firstcode[l]$.
3. For $l \leftarrow 1$ to *maxlength* do
   Set $nextcode[l] \leftarrow firstcode[l]$.
4. For $i \leftarrow 1$ to n do
   (a) Set $codeword[i] \leftarrow nextcode[l_i]$.
   (b) Set $symbol[l_i, nextcode[l_i] - firstcode[l_i]] \leftarrow i$.
   (c) Set $nextcode[l_i] \leftarrow nextcode[l_i] + 1$.
   The rightmost l_i bits of the integer $codeword[i]$ are the code for symbol i.

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/XC23Y

Decoding:

To decode a symbol represented in a canonical Huffman code,

1. Set $v \leftarrow nextinputbit()$.
   Set $l \leftarrow 1$.
2. While $v < firstcode[l]$ do
   (a) Set $v \leftarrow 2 * v + nextinputbit()$.
   (b) Set $l \leftarrow l + 1$.
   Integer v is now a legitimate code of l bits.
3. Return $symbol[l, v - firstcode[l]]$.
   This is the index of the decoded symbol.

The data integrity algorithm consists of two efficient cryptographic algorithms: ECC encryption encrypts the text file, and then Canonical Huffman encoding compresses the encrypted text file, which is subsequently reversed at the receiver side. This technique is best suited for situations in which the text file is encrypted once but must be decoded many times. The suggested approach is also applicable in resource-limited contexts.

5. DATA ANALYSIS

Network security has grown increasingly vital in recent decades, and intrusion detection systems play a crucial role in defending it. To identify intrusions, several machine learning algorithms have been utilized, with SVM being one of the most successful. In this study, we offer an effective intrusion detection methodology based on SVM with naive Bayes feature embedding. In particular, in the proposed techniques described in this research, the machine learning algorithm is used to distinguish between dangerous and normal network packets, which aid in collecting network packets and identifying which attack was performed on a specific network using rules from the KDD dataset. The machine learning-based classifiers identified incoming packets and discriminated between malicious and non-malicious network packets.

Dataset Description

In this research, we used the NSL-KDD dataset to demonstrate the superiority of the approach we proposed. The NSL-KDD dataset was upgraded from the KDD99 data sets generated by the Defense Advanced Research Projects Agency in MIT Lincoln's laboratories in the United States of America. At 78% and 75% datasets, there are a lot of repeated records in the KDD99 redundant training and test data, respectively. Redundant data sets might have a detrimental influence on the conclusion of assessment for much more accurate detection accuracy. The necessary KDD99 update was carried out, resulting in the new NSL-KDD datasets.

ISSN 1533-9211

DOI 10.17605/OSF.IO/XC23Y

Proposed Classification Algorithm

Support Vector Machine (SVM)

The SVM is a margin-based classification strategy that maximizes class separation while adhering to the concept of structural risk minimization. As a result, SVM has excellent generalization abilities and is resistant to overfitting. SVM may also handle non-linear classification issues by selecting kernel functions to transform the original feature space to alternative high-dimensional feature spaces where instances are linearly separable. SVM can also identify novelty. SVM is a supervised learning model containing learning algorithms that may be used for classification and regression analysis. In classification functions, SVM is used to discriminate between two classes in training data. The SVM's goal is to determine the optimal line for data separation. Support vectors are utilised to build models rather than using additional data. For linear and non-linear space, we employed the Gaussian kernel in SVM.

Naïve Bayesian Classifier (NBC)

The Naive Bayesian technique classifies attack packet types by computing the posterior probability. Based on the anomaly dictionary feature, Naive Bayesian is utilised to identify the anomaly as DoS, Remote to Local (R2L), User to Root (U2R), and probe. Naive Bayes and Naive Bayesian theorems can be represented mathematically in the following basic form:

$$Gain(D|B) = H(D) - \sum_{v \epsilon\, Values_B} \frac{|D_v|}{|D|}$$

The Proposed System using hybrid approach

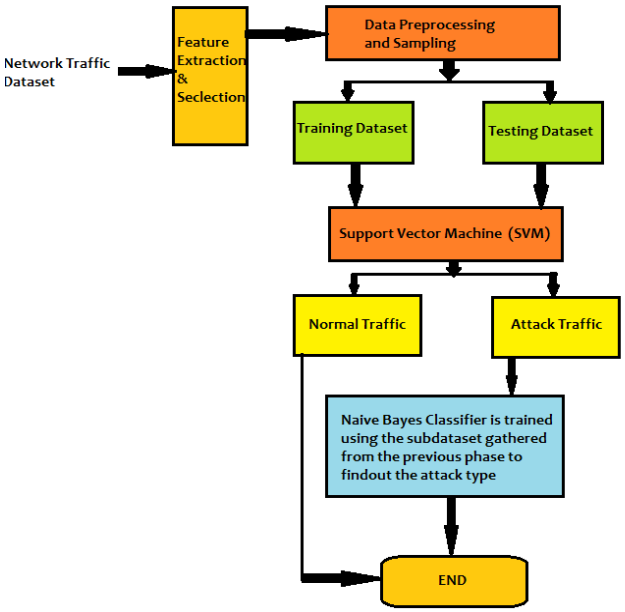
In the suggested system, the two algorithms are merged to provide an intrusion detection system. Fig 1 depicts the model of the proposed system. We filter the dataset using numerous cascade classifiers based on the features of the machine learning methods. Because the SVM method performs better on large datasets, it also performs better when employed as a binary classifier. As a consequence, Figure 1 demonstrates that the SVM algorithm is fed the entire dataset at first to filter out Normal and Attack traffic. Finally, the previously filtered dataset is utilised to train the Naive Bayesian classifier (NBC) to identify the kind of assault. The use of NBC is justified by the fact that it performs better with small datasets.

The primary purpose of using the SVM technique is to categorize and divide data into normal and attack instances. A support vector machine's main goal is to separate the provided data as best as feasible. The distance between the nearest spots after segregation is known as the margin. The method involves choosing a hyperplane with the greatest feasible margin between the support vectors in the provided data-sets.

ISSN 1533-9211

DOI 10.17605/OSF.IO/XC23Y

Fig 1: The Proposed Model



6. CONCLUSION

The real network should be supplied with a framework that will aid support the investigation procedure for an efficient network evaluation. When presenting it in court, using the information acquired within a network is highly unique. The previous system's limitation is that it uses a responsive examination approach that may not be capable of detecting the attack and demonstrating it successfully. The proactive method is used to make it more powerful. Once again, using an encryption mechanism ensures the integrity of the data. The proactive evaluation, characterisation, encryption, and pressure increase the structure's productivity.

In the previous year, several academics have been working on anomaly detection in a variety of domains. In this research, we explored the importance of identifying abnormalities in network packets. We developed a machine learning-based network anomaly detection system. We described existing anomaly detection methodologies in networks and defined anomaly detection successively. Notably, our technique can discover anomalous patterns that the trained model did not, implying promising findings for real-world Network Forensic applications. In this study, we suggested an efficient hybrid system dubbed SVMNB for Attack packet detection and kind of attack detection concurrently based on the given feature list to tackle certain concerns from earlier research. The system contemplates proactive network packet capture and application to an SVMNB-based algorithm for attack packet categorization. Furthermore, we used both actual and simulated data to compare our suggested strategy to other current techniques.

ISSN 1533-9211

DOI 10.17605/OSF.IO/XC23Y

References

1. M. Matsalu et al., “Digitaalse ekspertiisi to¨oj ¨ oupadevuse arendamine eesti kaitseliidu naitel,” Ph.D. dissertation, 2019.
2. Praveen P Naik, Prashantha S J.”An Approach for Building Intrusion Detection System by Using Data Mining Techniques ”International Journal of Emerging Engineering Research and Technology (IJEERT) Volume 2, Issue 2, May 2014, PP 112-118.
3. Amine Boukhtouta, Nour-Eddine Lakhdari,” Towards Fingerprinting Malicious Traffic", The 4th International Conference on Ambient Systems, Networks and Technologies (Science Direct).
4. David Mudzingwa and Rajeev Agrawal.” Evaluating Intrusion Detection and Prevention Systems Using Tomahawk and Wireshark”, Department of Electronics, Computer and Information Technology North Carolina A&T State University, Greensboro, NC, USA.
5. Mrs. GhatgeDipali D. – “Network Traffic Intrusion Detection System using Decision Tree & K-Means Clustering Algorithm” (IJETTCS) International Journal of Emerging Trends & Technology in Computer Science, Volume 2, Issue 5, September – October 2013.
6. T. Subbhulakshmi1, S. G. Keerthiga2 and R. Dharini3 – “Real-Time Intelligent Multilayer Attack Classification System” ICTACT Journal On Soft Computing, January 2014, Volume: 04, Issue: 02.
7. S. PraylaShyry, Efficient Identification of Bots by KMeans Clustering
8. S. Terry, B. Chow, 1999 DARPA Intrusion Detection Evaluation Data Set,http://www.ll.mit. edu/mission/ communications/cyber/CSTcorpora/ideval/data/1999data.html.
9. Ahmed, M., Samy, G., Maarop, N., Shanmugam, B., Magalingam, P. and Ahmad, R. Proposed Computer Forensic Approach for Cloud Computing Environment. Advanced Science Letters, 22(10), pp.3137-3141(2016).
10. Amran, A., Phan, R., Parish, D. and Whitley, J. Evidential structures and metrics for network forensics. International Journal of Internet Technology and Secured Transactions, 2(3/4), p.250(2010).
11. Avasthi, D. Network Forensic Analysis with Efficient Preservation for SYN Attack. IJCA, 24(2012).
12. Brown, R., & Pham, B. Image Mining and Retrieval Using Hierarchical Support Vector Machines. 11th International Multimedia Modelling Conference (IEEE), (2005).
13. Bartholomae, F. Cybercrime and cloud computing. A game theoretic network model. Managerial and Decision Economics, 39(3), pp.297–305 (2017).
14. Bass, T. Intrusion Detection Systems and Multisensor Data Fusion Communications of the ACM (2000).
15. B, C., V, K.K. and C, S.R. Comparative Study of cryptographic encryption algorithms. IOSR Journal of Electronics and Communication Engineering, 12(03), pp.66–71(2017).
16. Broucek, V. and Turner, P. Forensic Computing Developing a Conceptual Approach for an Emerging Academic Discipline, Australian security research Symposium (2001).
17. Castanedo, F. A Review of Data Fusion Techniques. The Scientific World Journal, pp.1–19(2013).
18. Dhammearatchi, D. Use of Network Forensic Mechanisms to Formulate Network Security. International Journal of Managing Information Technology, 7(4), pp.21–36 (2015).
19. Goel, R., Sardana, A. and Joshi, R. Wireless Honeypot: Framework, Architectures and Tools. International Journal of Network Security,
20. (2013).

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/XC23Y

21. He, J., Chang, C., He, P. and Pathan, M. Network Forensics Method Based on Evidence Graph and Vulnerability Reasoning. Future Internet, 8(4), p.54(2016).
22. Huang, J., Chen, Y., Ling, Z., Choo, K. and Fu, X. (n.d.). A Framework of Network Forensics and its Application of Locating Suspects in Wireless Crime Scene Investigation (2019)
23. Hikmatyar, M., Prayudi, Y. and Riadi, I. Network Forensics Framework Development using Interactive Planning Approach. International Journal of Computer Applications, 161(10), pp.41–48(2017).
24. ayamagarajothi, M. and Murugeswari, P. A Survey on Data Mining and Digital Forensics Techniques for Intrusion Detection and Protection System. International Journal of Advanced Research in Computer and Communication Engineering, 4(11), (2015).
25. Khobragade, P. and Malik, L. A Review on Data Generation for Digital Forensic Investigation using Data Mining. IJCAT International Journal of Computing and Technology, [online] 3(1), (2014).
26. Khurana, H., Basney, J., Mehedi,B., Freemon, M., Welch, V. and Y Butler Palantir: A Framework for Collaborative Incident Response and Investigation (2012).
27. Kim, J.-S., Kim, M. and Noh, B.-N. A Fuzzy Expert System for Network Forensics. Computational Science and Its Applications, ICCSA ,(2004).
28. Kumar, M.V. and Lalitha, D.T. Soft Computing: Fuzzy Logic Approach in Wireless Sensors Networks. Circuits and Systems, 07(08), pp.1242– 1249(2016).
29. Li, P., Wang, R., Zhang, Y. and Chen, D. A Network Forensics System Bypassing Web Local Encryption Businesses. Key Engineering Materials, 426-427, pp.494-498 (2010).
30. Liu, W., Zhang, L., Tao, D. and Cheng, J. Support vector machine active learning by Hessian regularization. Journal of Visual Communication and Image Representation, 49, pp.47–56(2017).
31. Md Siraj, M., Taha Albasheer, H.H. and Mat Din, M. Towards Predictive Real-time Multi-Sensors Intrusion Alert Correlation Framework. Indian Journal of Science and Technology, 8(12), (2015).
32. MbuguaChahira, J., KinanuKiruki, J. and KipronoKemei, P. A Proactive Approach in Network Forensic Investigation Process. International Journal of Computer Applications Technology and Research, 5(5), pp.304–311(2016).
33. Nikkel, B.J. A portable nek forensic evidence collector. Digital Investigation, 3(3), pp.127–135(2006).
34. Nirkhi, S.M. Data Mining : A Prospective Approach for Digital Forensics. International Journal of Data Mining & Knowledge Management Process, 2(6), pp.41–48(2012).
35. Perry, S. Network forensics and the inside job. Network Security, Science direct (12), pp.11–13(2006).
36. Pilli, E.S., Joshi, R.C. and Niyogi, R. Network forensic frameworks: Survey and research challenges. Digital Investigation, 7(1–2), pp.14– 27(2010).
37. Ponec, M., Giura, P., Wein, J. and Brönnimann, H. New payload attribution methods for network forensic investigations. ACM Transactions on Information and System Security, 13(2), pp.1–32(2010).
38. Rasmi, M. and Al Qerem, A. PNFEA: A Proposal Approach for Proactive Network Forensics Evidence Analysis to Resolve Cyber Crimes. International Journal of Computer Network and Information Security, 7(2), pp.25-32(2010)
39. Riadi, I., Eko, J. and Ashari, A. Internet Forensics Framework Based-on Clustering. International Journal of Advanced Computer Science and Applications, 4(12), (2013).
40. Rasmi, M. and Jantan, A. A New Algorithm to Estimate the Similarity between the Intentions of the Cyber Crimes for Network Forensics. Procedia Technology, 11, pp.540–547(2013).

The Seybold Report
ISSN 1533-9211
DOI 10.17605/OSF.IO/XC23Y

41. Reith, M., Carr, C. and Gunsch, G. An Examination of Digital Forensic Models. International Journal of Digital Evidence Fall, 1(3), (2002).
42. Ren, W. Modeling Network Forensics Behavior. Journal of Digital Forensic Practice, 1(1), pp.57–65(2006).
43. Saidi, A., Bendriss, E. and El Marraki, M. Implementation of a Mobile Agent System to Detect DoS/DDoS Flooding Attacks in Cloud Computing. International Journal of Security and Its Applications, 11(12), pp.35-44(2017).
44. Shah, M., Saleem, S. and Zulqarnain, R. Protecting Digital Evidence Integrity and Preserving Chain of Custody. The Journal of Digital Forensics, Security and Law, (2017)Shanmugasundaram, K., Memon, N., Savant, A. and Bronnimann, H. (n.d.). ForNet: A Distributed Forensics Network, LNCS (2003)
45. Tong, S. and Chang, E. Support vector machine active learning for image retrieval. Proceedings of the ninth ACM international conference on Multimedia - MULTIMEDIA '01 (2001).
46. Tian, J., Zhao, W., and DU, R. D-S Evidence Theory and its Data Fusion Application in Intrusion Detection - IEEE Conference Publication (2019).
47. Varshney, P.K. Multisensor data fusion. Electronics & Communication Engineering Journal, 9(6), pp.245–253(1997).
48. Wei, R. (n.d.). On A Reference Model of Distributed Cooperative Network Forensics System, Semantic Scholar (2019).
49. Wang, W. A graph oriented approach for network forensic analysis. Iowa State University Digital Repository (2010).
50. Yasinsac, A. and Manzano, Y. Honeytraps, A Network Forensic Tool. Multinational conference on systemic cybernetics and informatics (2002)
51. Yusoff, Y., Ismail, R. and Hassan, Z. Common Phases of Computer Forensics Investigation Models. International Journal of Computer Science and Information Technology, 3(3), pp.17–31(2011).
52. Zervas, E., Mpimpoudis, A., Anagnostopoulos, C., Sekkas, O. and Hadjiefthymiades, S. Multisensor data fusion for fire detection. Information Fusion, 12(3), pp.150–159(2011).
53. research.ijcaonline.org
54. docplayer.net
55. clok.uclan.ac.uk
56. repository.ihu.edu.gr
57. sce.zuj.edu.jo