ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse63942020.pdf
https://doi.org/10.30534/ijatcse/2020/63942020

Forensics in Private Cloud leveraging the techniques in Machine Learning

Savaridassan P.[1], Dr G. Maragatham[2]
[1]Department of Information Technology, SRM Institute of Science and Technology, INDIA, savaridp@srmist.edu.in
[2]Department of Information Technology, SRM Institute of Science and Technology, INDIA, maragatg@srmist.edu.in

ABSTRACT

Computerized crime scene investigation is a class of scientific science fascinate with the utilization of multifaceted data created, recorded and broadcast by different advanced gadgets as origin of proof in examinations and legitimate procedures. Computerized crime scene investigation can be separated to a many classes, for example, PC forensics, communication/network forensics, portable device forensics, private/internal cloud forensics, digital forensics. In recent times, private cloud service computing is developed as a well known computing service model in diverse requirements of the users. However in all aspects, internal cloud computing frameworks need support for digital forensic examinations. The primary objective of computerized forensics is to demonstrate the evidence of specific information within the available digital source. This overall paper introduces a thorough study of different systems and proposals with respect to various classes in computerized crime scene investigation with an emphasis on private/internal cloud computing forensics. Classic Digital criminology classes, their systems, limitations in the systems and proposals were examined. At this point the methodological perspective and existing difficulties of cloud crime scene investigation were focused. Also, the comparison based on various parameters, outlined about disadvantages, differences and similarity of a many proposed internal cloud computing systems/structures and also to provide future research trends.

Key words: Assessment phase, Evidence, Internal cloud computing forensics, Investigation method, openstack cloud forensics

1. INTRODUCTION

Cloud forensics has been an area of security where different techniques employed to gather evidence from the victim's machine. This is due to the fact that the target from which the result has to be acquired is virtual and the physical hardware is located at the remote location. In research published in 2013 by Josiah Dykstra and Alan T. Sherman [2], a forensic tool was introduced which is helpful in gathering the evidence from the instance. This tool could be integrated with OpenStack to gather the logs of Nova, the host firewall and the disk image. It could additionally be adopted for live monitoring, capturing metrics, and auditing. Recently Openstack has introduced some of these features in their new upgrades. Josiah Dykstra and Alan T. Sherman[4] also published the approaches that can be followed to gather and analyze the evidence from a public cloud(Iaas). The approach was centered around Amazon Ec2. Amazon Ec2 being a proprietary technology offers a lot of difficulties while gathering the evidence. The authors also assessed the performance of forensic tools such as Encase and AccessData FTK. Now the question arises that though there are many tools readily available in the market which are reliable for the investigation, then why a number of cases reported where the criminals are actually prosecuted is so less? Emi Morioka and Mehrdad S. Sharbaf [5] published a research in 2016 where they addressed these issues more clearly, the also mentioned the challenges identified by the NIST working group. Issues such as type of architecture compromised, anti-forensics, legal jurisdictions under which the suspect is to be prosecuted were highlighted. They also shed light on scenarios of a multi-tenancy environment where they expressed the difficulty to access the instance of one tenant out of many other tenants which could lead to compromising the confidentiality of other tenants. Other than the challenges identified by NIST, [3] ACPO principles and guidelines can also act as bottlenecks during the investigation. The investigator is expected to withhold these principles. Apart from being the bottlenecks they also guide the methods and ways in which the seized evidence is to be dealt with. In all the above mentioned published researches ([2],[4],[3],[5]), the ease of acquiring the evidence has always been the aim. But then what about identifying the evidence? How to know where to look? Changwei Liu, Anoop Singhal, and Duminda Wijesker in 2017[1] introduced a prolog based tool in their research that could relate the pieces of evidence and recreate the whole flow of the scenario via a logic evidence graph. The data gathered as evidence was classified into three categories facts, rules, and consequences. Openstack was used to deploy private cloud (same as in [2]). Very few of the researches [8] have utilized the effectiveness of machine learning and neural net in the field of forensics, as most of them were focused upon evidence gathering. We will be using Openstack to deploy our private cloud (as in [1]). Our aim through this research is to use the machine learning approach to support the process of investigation along with trying to ease the process of gathering the data as evidence from the deployed instance before it is removed. Our work will mainly be focused upon private clouds deployed by Openstack in order to avoid any legal issues due to proprietary technology.

Here is a list of challenges and scenarios that occur during evidence gathering or investigation in forensics:

1. In many scenarios where the attackers were successful in obtaining complete access over the instance, any interaction with the instance such as login could lead to exposing the credentials to the attacker. In the work published by Abdullah and Mohamed, they state that, IDS systems encounters challenges at transforming the deployment from legacy network to cloud computing support design [6].

2. In such cases, the admins generally try to delete the instance which leads to a dead end.

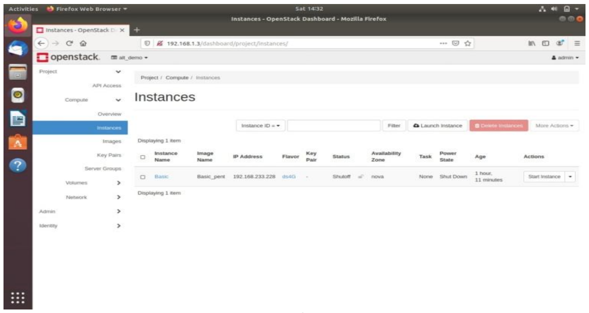
3. Using rule-based tools for forensics many known causes can be identified, but at the same time, many unknown cases can be skipped.

2. SYSTEM OVERVIEW

Many of the security specialists would concede to the explanation that it is difficult to construct a totally secure framework. In this approach it is imperative to have great resistance systems to remain alert for threats. In this area, we examine cutting edge approaches and give an outline of the potential outcomes to safeguard a system. System based defense techniques are a significant countermeasure since they include an extra layer of security to the system and in this manner can limit the danger of the threats and attacks. Intrusion identification frameworks are considered as a proper answer for information assurance. These frameworks are intended to effectively distinguish and conceivably prevent intruders by observing various sources like particularly the system traffic. There are various ways to deal with how the attackers are identified. Different approaches utilize mostly the static anomaly detection; others techniques identify deviations from usual traffic (anomaly recognition). Over the previous years, numerous new discovery strategies have been created. One kind of these new methodologies is leveraging the Artificial Intelligence and Machine learning approach and adopts the statistical models to identify and detect the intrusions.

2.1 Environment Set up

We will be using an ubuntu bionic virtual machine deploy OpenStack which can be used to create an instance to host a vulnerable website hosted on a word-press server with IP [192.168.233.228]. We will be using Kali Linux running on a separate physical machine connected to the network with IP 192.168.1.2 communicating with the server via port forwarding. As an attacker, we attempted to gain root access to the server and remove the current user. The packet sniffer installed on the Ubuntu virtual machine was used to capture the packets which will later be used as the evidence during the investigation. We also are collected the log files of both OpenStack and the server to identify the malicious activity and also to support our investigation. Figure 1, depicts the usable view of the private cloud navigation pane.



Displaying 1 item

☐	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
☐	Basic	Basic_pent	192.168.233.228	ds4G	-	Shutoff	nova	None	Shut Down	1 hour, 11 minutes	Start Instance

Displaying 1 item

Figure 1: Screenshot of Openstack dashboard

2.2 Malicious Packet classifier

Intrusion and Detection approaches (IDS) are available as hardware products or softwares process that screen PC systems or the network packet traffic and search for malicious behaviors. If there arise an occurrence of an attack incident or violation of the policy, the monito system creates reports for the administration. Intrusion and detection techniques are essentially utilized to perform the monitoring of the underlying system activities. In our experiment, we create a classifier to classify malicious traffic from the rest of the traffic. We have used our own dataset along with the CTU-13 dataset[9] for training. The model was built using R and deployed on a shiny server.

2.3 Memory Forensics

In 2018, Chang Liu ,Wei Song and others[8] published a research where they used neural graphs to process the memory images taken from the infected hosts and detect kernel object -

manipulation. The results were proven to be very efficient compared to other approaches.

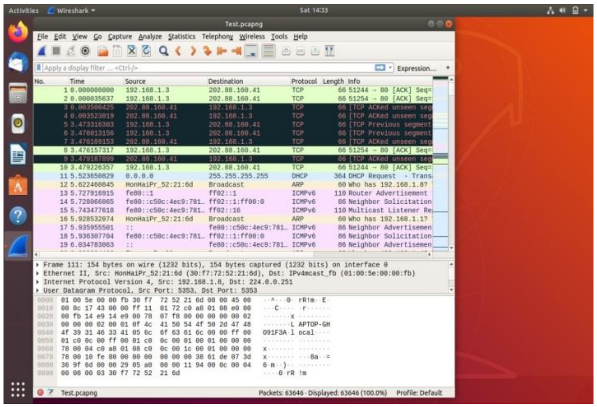
- It doesn't depend on the information on working framework source code or internal operating system information structures.
- Can default produce important features of kernel specific objects derived from the plain bytes in memory dump without the manual input from the expert system analysis.
- It adopts the deep learning neural system models for proficient parallel operations and calculations.
- It derives relative features that can be utilised for defend against the attacks and threats like tag manipulaton, DKOM technique like hide running process.

3. METHODOLOGY

This section presents the pre-investigation and post-investigation phases. In the pre-investigation phase, we focus on the evidence gather ing from the victim's machine followed by a post-investigation phase where the evidence gathered is cleaned and then sent to the classifier where the malicious traffic is classified and visualizations are performed.

3.1 Pre Investigation Phase

This phase focuses on evidence gathering. The main sources of evidence in our experiment are log files of openstack and server, as well as the packets collected through wireshark(or any other sniffer tool). Figure 2, illustrates the packets captured contains the features which are extracted using CICFlowMeter.


Activities
Wireshark
Sat 14:33
Test.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...

```
No.   Time         Source                 Destination            Protocol Length Info
    1 0.000000000  192.168.1.3            202.88.160.41          TCP        66 51244 → 80 [ACK] Seq=
    2 0.000035637  192.168.1.3            202.88.160.41          TCP        66 51254 → 80 [ACK] Seq=
    3 0.003500425  202.88.160.41          192.168.1.3            TCP        66 [TCP ACKed unseen seg
    4 0.003523019  202.88.160.41          192.168.1.3            TCP        66 [TCP ACKed unseen seg
    5 3.473316303  192.168.1.3            202.88.160.41          TCP        66 [TCP Previous segment
    6 3.476013156  192.168.1.3            202.88.160.41          TCP        66 [TCP Previous segment
    7 3.476109153  202.88.160.41          192.168.1.3            TCP        66 [TCP ACKed unseen seg
    8 3.476157317  192.168.1.3            202.88.160.41          TCP        66 51254 → 80 [ACK] Seq=
    9 3.479187899  202.88.160.41          192.168.1.3            TCP        66 [TCP ACKed unseen seg
   10 3.479226357  192.168.1.3            202.88.160.41          TCP        66 51244 → 80 [ACK] Seq=
   11 5.523650829  0.0.0.0                255.255.255.255        DHCP      364 DHCP Request  - Trans
   12 5.622460845  HonHaiPr_52:21:6d      Broadcast              ARP        60 Who has 192.168.1.8?
   13 5.727916915  fe80::1                ff02::1                ICMPv6    110 Router Advertisement
   14 5.728066065  fe80::c50c:4ec9:781…   ff02::1:ff00:0         ICMPv6     86 Neighbor Solicitation
   15 5.743477618  fe80::c50c:4ec9:781…   ff02::16               ICMPv6    110 Multicast Listener Re
   16 5.928532074  HonHaiPr_52:21:6d      Broadcast              ARP        60 Who has 192.168.1.1?
   17 5.935955501  ::                     fe80::c50c:4ec9:781…   ICMPv6     86 Neighbor Advertisemen
   18 5.936307704  fe80::c50c:4ec9:781…   ff02::1:ff00:0         ICMPv6     86 Neighbor Solicitation
   19 6.034783063  ::                     fe80::c50c:4ec9:781…   ICMPv6     86 Neighbor Advertisemen

▸ Frame 111: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits) on interface 0
▸ Ethernet II, Src: HonHaiPr_52:21:6d (30:f7:72:52:21:6d), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
▸ Internet Protocol Version 4, Src: 192.168.1.8, Dst: 224.0.0.251
▸ User Datagram Protocol, Src Port: 5353, Dst Port: 5353

0000  01 00 5e 00 00 fb 30 f7  72 52 21 6d 08 00 45 00   ··^···0· rR!m··E·
0010  00 8c 17 43 00 00 ff 11  01 72 c0 a8 01 08 e0 00   ···C···· ·r······
0020  00 fb 14 e9 14 e9 00 78  07 f8 00 00 00 00 00 02   ·······x ········
0030  00 00 00 02 00 01 0f 4c  41 50 54 4f 50 2d 47 48   ·······L APTOP-GH
0040  4f 39 31 46 33 41 05 6c  6f 63 61 6c 00 00 ff 00   O91F3A·l ocal····
0050  01 c0 0c 00 ff 00 01 c0  0c 00 01 00 01 00 00 00   ········ ········
0060  78 00 04 c0 a8 01 08 c0  0c 00 1c 00 01 00 00 00   x······· ········
0070  78 00 10 fe 80 00 00 00  00 00 00 38 61 de 07 3d   x······· ···8a··=
0080  36 9f 6d 00 00 29 05 a0  00 00 11 94 00 0c 00 04   6·m··)·· ········
0090  00 08 00 03 30 f7 72 52  21 6d                     ····0·rR !m

Test.pcapng          Packets: 63646 · Displayed: 63646 (100.0%)     Profile: Default
```

Figure 2: Screenshot of Packets captured

3.2 Post Investigation Phase

This phase consists of different steps mentioned as follows:

- Passing the data into the trained model to classify malicious packets. The identified packets are then passed on for visualization along with their timestamp to relate them with the activity via log files.
- For log classification, we used unsupervised learning with data visualization to have a better understanding of the scenario.

3.3. Classification Model

The neural net model consists of 4 layers out of which 2 are hidden layers. The input layer is designed to accept 37 parameters. There are five levels of classifications that are taken into consideration. Our own generated dataset along with CTU-13 dataset [9] was used to train the neural net. Figure 3 represents the neural network architecture and the flow.

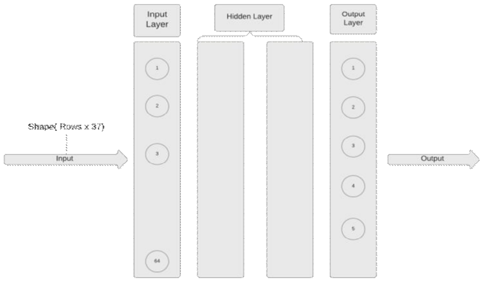


Figure 3: Diagram depicting the layers of neural net classifier

The parameter selection was done on the basis of correlation between the attributes. Highly correlated features were removed and random forest was used to cross check the result obtained. Numbers of parameters were reduced from 78 to 37. Parameters such as 'TCP Payload' were excluded from the set, because of their contribution to false positives in favor of malicious packets. The Figure 4, portrays the correlation metrics among the data features selected from the network packets.

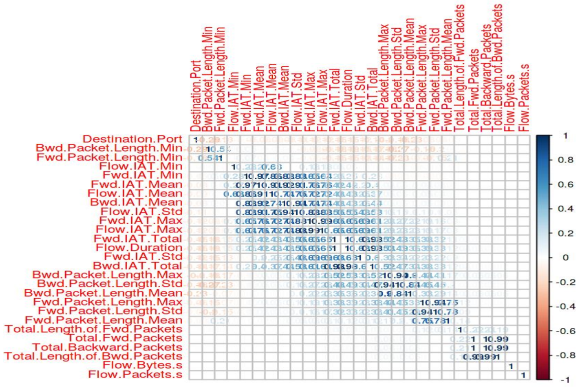


Figure 4: Correlation plot between features extracted from packets

3.4. Log Analysis

Logs generated during the whole experiment is quite large, in order to reduce the efforts we took the timestamp of the malicious packets into consideration. The prototype developed will be visualizing the requests made to the server, as well as the type of content accessed by the clients which can be traced through the time. Figure 5, represents the visualization of requests made to the server via log data collected from server.

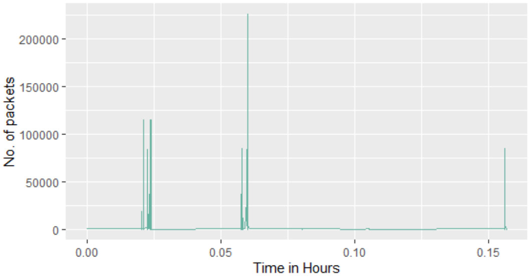


Figure 5: Time series graph for packets transaction between server and client

The suspicious log data based on time stamp is mapped onto tokens. These tokens are then arranged in a sequential manner and then will be used as an input for k-means algorithm. The clusters formed will give a clear representation of the operations performed during that duration, such as changing access permissions or uploading a malicious code to the server as an add-on.

4. CONCLUSION AND FUTURE WORK

The prototype developed to support our work demonstrated use of different techniques for forensic investigation. Neural net classifiers performed really well with an accuracy of 91% on the test data set and 89% on a real scenario created during the experiment. The time taken to classify the malicious packets from the rest is something that makes it unsuitable for real-time. Apart from the neural net, we tried different machine learning algorithms such as random forest, k-nearest neighbors, support vector machine enabled by many providers as listed in the work by syed and vijaya [7] which lead to over-fitting due to the nature of the data we decided to used. To implement these algorithms we used the “WEKA” tool to create the models and then test them, and later we used Kaggle notebook to train our neural net model.

For log analysis, we laid a lot of emphasis on data visualization because of the volume of data generated during our experiment. In real-world scenarios, visualization will make the understanding of the data much easier compared to techniques such as natural language processing. For future work, we look forward to introducing natural language processing to observe the results.

This work is a baseline for future research where researchers can extend and develop a framework for packet dissection or malware detection, or predict the type of request the client is trying to make and prepare to handle it beforehand.

REFERENCES

[1] Changwei Liu, Anoop Singhal and Duminda Wijeseker , **identifying evidence for cloud forensic analysis**, IFIP International Conference on Digital Forensics, 2017

[2] Josiah Dykstra, Alan T. Sherman, **Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform,** (Cyber Defense Lab, Department of CSEE, University of Maryland, Baltimore County (UMBC), 1000 Hilltop Circle, Baltimore, MD 21250, United States), 2013

[3] Denis Reilly, Chris Wren and Tom Berry, **Cloud Computing: Pros and Cons for Computer Forensic Investigations**, International Journal Multimedia and Image Processing (IJMIP), Volume 1, Issues 1/2, 2011

[4] Josiah Dykstra and Alan T. Sherman , **Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques,** Digital Investigation, Elsevier 2012
https://doi.org/10.1016/j.diin.2012.05.001

[5] Emi Morioka and Mehrdad S. Sharbaf, **Digital Forensics Research on Cloud Computing: An investigation of Cloud Forensics Solutions**, IEEE Symposium on Technologies for Homeland Security (HST), 2016
https://doi.org/10.1109/THS.2016.7568909

[6] Abdallah Ghourabi, Mohamed Jelidi, **Experimental Evaluation of a Hybrid Intrusion Detection System,** International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.6, November – December 2019
https://doi.org/10.30534/ijatcse/2019/65862019

[7] Syed.Karimunnisa, Dr.Vijaya Sri Kompalli, **Cloud Computing: Review on Recent Research Progress and Issues,** International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.2, March - April 2019
https://doi.org/10.30534/ijatcse/2019/18822019

[8] Nickson K Marie, Victor R Kebande and H. S. Venter , **Diverging deep learning cognitive computing techniques into cyber forensics**, Forensic science International: synergy, 2019
https://doi.org/10.1016/j.fsisyn.2019.03.006

[9] S.Garcíaab, M.Grillb, J.Stiborekb, A.Zuninoa, **An empirical comparison of botnet detection methods**, Computers & Security, Elsevier, 2014
https://doi.org/10.1016/j.cose.2014.05.011

[10] Wei Song , Heng Yin , Dawn Song , Chang Liu, **DeepMem: Learning Graph Neural Network Models for Fast and Robust Memory Forensic Analysis**, Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications SecurityJanuary 2018 Pages 606–618, 2018

[11] L. Daniel, **Digital forensics for legal professionals: understanding digital evidence from the warrant to the courtroom**. Elsevier, 2011.

[12] C. Hargreaves and J. Patterson, **An automated timeline reconstruction approach for digital forensic investigations**,Digital Investigation, vol. 9, pp. S69–S79, 2012.

[13] M. Reith, C. Carr, and G. Gunsch, **—An examination of digital forensic models international journal of digital evidence**, 2002.

[14] M. D. Kohn, M. M. Eloff, and J. H. Eloff, **–Integrated digital forensic process model**,‖ Computers & Security, vol. 38, pp. 103–115, 2013.

[15] F. Servida and E. Casey, **–Iot forensic challenges and opportunities for digital traces**,‖ Digital Investigation, vol. 28, pp. S22–S29, 2019. https://doi.org/10.1016/j.diin.2019.01.012

[16] E. Casey, **Digital evidence and computer crime: Forensic science, computers, and the internet.** Academic press, 2011.

[17] R. McKemmish, **What is forensic computing?** Australian Institute of Criminology Canberra, 1999.

[18] K. Kent, S. Chevalier, T. Grance, and H. Dang, **–Guide to integrating forensic techniques into incident response**,‖ NIST Special Publication, vol. 10, no. 14, pp. 800–86, 2006.

[19] J. R. Vacca, **Computer Forensics: Computer Crime Scene Investigation (Networking Series)**, (Networking Series). Charles River Media, Inc., 2005.

[20] E. S. Pilli, R. C. Joshi, and R. Niyogi, **–Network forensic frameworks: Survey and research challenges,‖ digital investigation,** vol. 7, no. 1-2, pp. 14–27, 2010.

[21] M. Köhn, M. S. Olivier, and J. H. Eloff, **–Framework for a digital forensic investigation.‖ in ISSA**, 2006, pp. 1–7.

[22] G. Palmer, **–A road map for digital forensics research-report from the first digital forensics research workshop** (dfrws),‖ Utica, New York, 2001.

[23] B. Hitchcock, N.-A. Le-Khac, and M. Scanlon, **–Tiered forensic methodology model for digital field triage by non-digital evidence specialists,‖ Digital investigation,** vol. 16, pp. S75–S85, 2016. https://doi.org/10.1016/j.diin.2016.01.010

[24] M. Rogers, Dcsa: **Applied digital crime scene analysis,‖ Tipton & Krause**, 2006.

[25] S. Von Solms, C. Louwrens, C. Reekie, and T. Grobler, **A control framework for digital forensics,‖ in IFIP International Conference on Digital Forensics**. Springer, 2006, pp. 343–355. https://doi.org/10.1007/0-387-36891-4_27

[26] H. C. Lee, T. Palmbach, and M. T. Miller, Henry Lee's **crime scene handbook**. Academic Press, 2001.

[27] S. Ó. Ciardhuáin, **An extended model of cybercrime investigations**,‖ International Journal of Digital Evidence, vol. 3, no. 1, pp. 1–22, 2004.

[28] B. Carrier and E. H. Spafford, **–An event-based digital forensic investigation framework**,‖ in Digital forensic research workshop, 2004, pp. 11–13.

[29] V. Baryamureeba and F. Tushabe, **―The enhanced digital investigation process model**,‖ in Proceedings of the Fourth Digital Forensic Research Workshop, 2004, pp. 1–9.

[30] M. K. Rogers, J. Goldman, R. Mislan, T. Wedge, and S. Debrota,**–Computer forensics field triage process model**,‖ Journal of Digital Forensics, Security and Law, vol. 1, no. 2, p. 2, 2006.

[31] S. A. Ali, S. Memon, and F. Sahito, **–Challenges and solutions in cloud forensics**,‖ in Proceedings of the 2018 2nd International Conference on Cloud and Big Data Computing. ACM, 2018, pp. 6–10.

[32] M. Mabey, A. Doupé, Z. Zhao, and G.-J. Ahn, **–Challenges, opportunities and a framework for web environment forensics**,‖ in IFIP International Conference on Digital Forensics. Springer, 2018, pp. 11– 33. https://doi.org/10.1007/978-3-319-99277-8_2

[33] S. Raghavan, **–Digital forensic research: current state of the art**,‖ CSI Transactions on ICT, vol. 1, no. 1, pp. 91–114, 2013.

[34] J. Yadav, **–The impact of digital forensics in future**,‖ 03 2017.

[35] A. Agarwal, M. Gupta, S. Gupta, and S. C. Gupta, **–Systematic digital forensic investigation model**,‖ International Journal of Computer Science and Security (IJCSS), vol. 5, no. 1, pp. 118–131, 2011.

[36] B. Carrier, E. H. Spafford et al., **–Getting physical with the digital investigation process**,‖ International Journal of digital evidence, vol. 2, no. 2, pp. 1–20, 2003.

[37] R. Van Baar, H. Van Beek, and E. Van Eijk, **–Digital forensics as a service: A game changer**,‖ Digital Investigation, vol. 11, pp. S54–S62, 2014.

[38] B. Martini and K.-K. R. Choo, **–An integrated conceptual digital forensic framework for cloud computing**,‖ Digital Investigation, vol. 9, no. 2, pp. 71–80, 2012.

[39] D. Quick and K.-K. R. Choo, **―Data reduction and data mining framework for digital forensic evidence: storage, intelligence, review and archive,‖ Trends & Issues in Crime and Criminal Justice**, vol. 480, pp. 1–11, 2014.

[40] A. Al-Dhaqm, S. Razak, S. H. Othman, K.-K. R. Choo, W. B. Glisson,A. Ali, and M. Abrar, –Cdbfip: **Common database forensic investigation processes for internet of things**,‖ IEEE Access, vol. 5, pp. 24 401– 24 416, 2017.

[41] E. Benkhelifa, B. E. Thomas, Y. Jararweh et al., **–Framework for mobile devices analysis,‖ Procedia Computer Science**, vol. 83, pp. 1188–1193, 2016.

[42] M. Petraityte, A. Dehghantanha, and G. Epiphaniou, **–Mobile phone forensics: an investigative framework based on user impulsivity and secure collaboration errors,‖ in Contemporary Digital Forensic Investi- gations of Cloud and Mobile Applications**. Elsevier, 2017, pp. 79–89.

[43] H. Dreger, A. Feldmann, M. Mai, V. Paxson, and R. Sommer, **–Dynamic application-layer protocol analysis for network intrusion detection,‖ in 15th USENIX security symposium. USENIX Association**, 2006, pp. 257–272.

[44] G. Maier, R. Sommer, H. Dreger, A. Feldmann, V. Paxson, and F. Schneider, **–Enriching network security analysis with time travel**,‖ in ACM SIGCOMM Computer Communication Review, vol. 38, no. 4. ACM, 2008, pp. 183–194.

[45] M. Saadeh, A. Sleit, M. Qatawneh, and W. Almobaideen, –**Authentication techniques for the internet of things: A survey**,‖ in 2016 Cybersecurity and Cyberforensics Conference (CCC). IEEE, 2016, pp. 28–34.
https://doi.org/10.1109/CCC.2016.22

[46] K. Shanmugasundaram, N. Memon, A. Savant, and H. Bronnimann, –**Fornet: A distributed forensics network**,‖ in International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security. Springer, 2003, pp. 1–16.

[47] W. Wang and T. E. Daniels, – **A graph based approach toward network forensics analysis**,‖ ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 1, p. 4, 2008.

[48] T. V. Lillard, **Digital forensics for network, Internet, and cloud computing: a forensic evidence guide for moving targets and data**. Syngress Publishing, 2010.

[49] A. Singhal, C. Liu, and D. Wijesekara, –**Poster: A logic based network forensics model for evidence analysis**,‖ in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. ACM, 2015, pp. 1677–1677.

[50] M. Neugschwandtner, P. M. Comparetti, G. Jacob, and C. Kruegel, –**Forecast: skimming off the malware cream**,‖ in Proceedings of the 27th Annual Computer Security Applications Conference. ACM, 2011, pp. 11–20.

[51] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel, and E. Kirda, –**Scalable, behavior-based malware clustering**.‖ in NDSS, vol. 9. Cite- seer, 2009, pp.8–11.

[52] T. Tafazzoli, E. Salahi, and H. Gharaee, —**A proposed architecture for network forensic system in large-scale networks,**‖ arXiv preprint arXiv:1508.01890, 2015.

[53] L. Jiang, G. Tian, and S. Zhu, –**Design and implementation of network forensic system based on intrusion detection analysis**,‖ in 2012 International Conference on Control Engineering and Communication Technology. IEEE, 2012, pp. 689–692.

[54] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, –**Internet of things forensics: Challenges and approaches**,‖ in 9th IEEE International Conference on Collaborative computing: networking, Applications and Worksharing. IEEE, 2013, pp. 608–615.

[55] M. H. Qasem and M. Qatawneh, **Parallel hill cipher encryption algorithm**,‖ International Journal of Computer Applications, vol. 179, no. 19, pp. 16–24, 2018.
https://doi.org/10.5120/ijca2018916326