



An Efficient Approach of Threat Hunting Using Memory Forensics

Danish Javeed¹, Muhammad Taimoor Khan², Ijaz Ahmad³, Tahir Iqbal⁴, Umar Mohammed Badamasi⁵, Cosmas Obiora Ndubuisi⁶ and Aliyu Umar⁷

^{1,4} Northeastern University, Shenyang, Liaoning province, China

² Riphah Institute of Science and Technology, Islamabad, Pakistan

^{3,5,6,7} Changchun University of Science and Technology, China

¹*thedanishkhn@gmail.com*

ABSTRACT

The capacity and occurrence of new cyber-attacks have shattered in recent years. Such measures have very complicated workflows and comprise multiple illegal actors and organizations. Threat hunting demonstrates the process of proactively searching through networks for threats based on zero-day attacks by repeating the hunting process again and again. Unlike threat intelligence, it uses different automated security tools to collect logs in order to provide a pattern for making new intelligence-based tools by following those logs. According to our research findings about “threat hunting tools” there’s a major flaw that the designed tools are limited to the collection of logs. It works completely on logs for generating new patterns avoiding system’s main memory. Codes written directly to memory fail this process to provide proactive hunting. To overcome this major challenge, we are proposing two distinct methods, either by generating malicious code alerts or by binding memory forensics processes with threat hunting tools to make active hunting possible.

Keywords: Information Security, Memory Forensics, Threat Hunting, Logs, Threat Intelligence, Automated Tools.

1 INTRODUCTION

Proactive Threat Hunting is the practice of proactively searching around the networks or datasets for the detection and responding to progressive cyber threats that escape outdated rule- or signature-based security panels. Threat hunting incorporates the use of threat intelligence, analytics, and automated security tools with experience, human intelligence and some skills.

Cyber security comprises innovations, procedures and controls which are planned to guarantee frameworks, systems and info from cyber-attacks. Feasible cyber security condenses the risk of cyber-attacks, and guards’ associations and folks from the Illegitimate misuse of frameworks, systems and developments. The bigger part of cyber-attacks is automated and erratic, abusing known vulnerabilities. Instead of concentrating on specific associations, your association might be cracked at the present-day and you won’t be aware of it.

The application of cyber security could be helpful in order to forestall cyber-attacks, info breaches and data scam and it can help in chance administration. At that point where an organization has a very strong feeling of network security and a feasible incidence reaction design, it’s better prepared to forestall and diminish these attacks. For example, end Client insurance data protection and gatekeepers against bad luck or break-in while similarly investigating PCs for pernicious code. Cybercrime includes a PC and a network. Sometimes, the PC may have been utilized as a part of request to perpetrate the wrongdoing, and in different cases, the PC may have been the objective of the wrongdoing. Offenses that are committed against individuals or meetings of people with an illegal thought course to persistently hurt the notoriety of the fatality or effect physical or mental harm, or misfortune, to the victim directly or in an indirect way, developing current media transmission networks, for example, Internet (networks including yet not constrained to Chat

rooms, mails, see sheets and meetings) and cell phones. Cybercrime may intimidate a man or a country's safety and money associated welfare. Threat hunting is a pre-emptive and monotonous technique to identifying malicious activities. On the Descending Gage of Cyber Security, hunting falls in the dynamic defense group as it is performed principally by a human specialist. Though threat hunters have to depend on deeply on mechanization and machine support, the procedure itself can't be completely automatic nor can any invention accomplish hunting for an expert. One of the human's important aids to any hunting is the early commencement of what sort of threat the analyst would want to hunt and in what way the analyst may discover that type of malevolent movement in the System. We naturally state to this early formation as the hunt's hypothesis, however it's certainly just a declaration about the hunter's testable thoughts of what threats might be in the system and in what way to find those threat. There are two main components to produce any of hunting hypotheses.

First, an analyst's aptitude to make hypotheses is derived from the observations. An observation might be as simple as observing a specific incident that "just doesn't seem right" or something more complex, such as a assumption about current threat actor movement in the system grounded on an amalgamation of previous knowledge with the actor and exterior threat intelligence. The 2nd thought to understand is that hypotheses must have to be testable i.e. it must be something that you have a slight chance of finding in the data to which you have access. Good hunts rely on the hunter's talent to distinguish what data and tools are necessary to test the hypotheses. To completely test hypotheses, it also demands the correct analysis tools and techniques that can concurrently take benefit of info from the system as well as adversaries. A good threat-hunting platform supports analysts in creating hypotheses and decreases fences to testing those hypotheses by providing ready access to the data and tools required to accomplish the tests. Threat hunting is appropriately centered on threats. Furthermore, to be a threat, an enemy must have three things: the expectation, ability and chance to do hurt. Threat hunters center their inquiry on enemies who have those three qualities and who are as of now inside the networks and frameworks of the threat seekers' association, where they have expert to gather information and convey countermeasures. Cyber threat seekers work with a wide range of security observing arrangements, for example, firewalls, antivirus programming, network security checking, information misfortune prevention, network interruption recognition, insider threat

identification, and other security instruments. Other than checking the network at the hierarchical level, they likewise look at endpoint information. They accumulate occasion logs from whatever number places as could reasonably be expected, as their work requires an adequate measure of security information.

Threat hunting is a proactive way to deal with distinguishing foes as opposed to responsively sitting tight for an alarm to go off. Most associations are doing threat hunting to a few degrees today. Their very own comprehension hunting development will encourage control associations that need to develop these abilities. By then, associations need to engage threat investigators with the correct preparing, datasets and mechanized stages to move toward becoming examination driven safeguards. Threat hunting is aimed at discovering abnormal activities that otherwise can result in grave damage to your company. Understanding of normal activities in your environment is a prerequisite to comprehending activities that are not normal. If you understand normal operational activities, then anything abnormal should stand out and be noticed. A good threat-hunting practice requires threat hunters think like an attacker. Normally, the task of threat hunters is to chase adversaries proactively and put an end to the chance of intrusions. If the attack has been taken place, however, they need to mitigate its impact in order to reduce damage. However, always looking for the signs of intrusion is not a very good approach. Rather, threat hunters should work to anticipate an attacker's next move. OODA is an abbreviation of Observe, Orient, Decide and Act. Military personnel apply OODA when they carry out combat operations. Likewise, threat hunters use OODA during cyber warfare. In the context of threat hunting, OODA works as:

Observe: A first phase that involves routine data collection from endpoints, Orient: Understanding the collected data thoroughly and combining this information with other collected information to help understand its meaning. After that, analyzing whether the sign of Command & Control (C&C) over traffic occurs or any sign of attack is detected, Decide: Once you have analyzed the information, then you need to identify the course of action. If the incident occurs, threat hunters will execute the incident response strategy, Act: The last phase involves the execution of the plan to put an end to the intrusion and enhance the company's security posture. Further measures are taken to prevent the same type of attack in the future.

2 RELATED WORK

This paper discusses that threat from insiders is a problem which is not going away, and all of the implications shows that it is getting worse day by day. An effective insider threat justification approach will help diminish risk, as well as financial, lawful and in addition the potential destruction to the association. [4] This paper describes expending analytics and machine learning; specialists generate and work from starting point that helps recognize fresh and irregular patterns for quicker recognition and remediation of iden-tified and unidentified threats. [14] his paper Give us a complete overview of TC and also the extensibility of the platform to enable users to adapt and create automation for their processes, rather than forcing them to adapt their processes to ThreatConnect's paradigm. [3]. this paper lab-els two important components for producing hunting hypotheses: They are mainly grounded on observations, and they must be testable. First, an analyst's aptitude to generate hypotheses is obtained from observations. An observation might be as artless as observing a specific event that "just doesn't seem right" or somewhat more complex, such as an actor activity is running in the environment and we just assume it and this assumption is based on the past experience with the actor and some external threat intelligence. The second concept is that once there is a hypothesis it must be testable. That is, there should be something about your hypothesis that you may have at least a slight chance of finding it in the data to which you have access. This paper gives a complete survey and key results (statistics) SANS 2017 Threat Hunting Survey show that, for numerous organizations, it is still fresh and poorly defend from a course and organizational perspective. Unluckily, there are still numerous organizations that are still reacting to incidents which already cause the damage instead of proactively looking for those threats inside the system. But there is one thing which should be kept in mind that threat hunting cannot be completely automated. [6]. the business context is almost organization knowledge and its technical context is the footprint of malicious activities inside the organization. This paper aims that how a technical information and business information highlight the threat information. Most Threat feed consist of IOC. [7] To compound the issue the threats which exist inside within the organization creates a big problem for the passive defenders. And that is the reason that Passive defense strategies are no longer feasible for hunting the attackers. One of the major breaches of confidential information was carried out by an insider in the US. The detection delta rate

is still high i-r 90days approx. Threat hunters must focus on going after, or hunting, the humans by simply shifting through logs and changes may be operational, but it does not lend to a proactive pursuit of intrusion within or against an organization. Asking a possible hunting provider the right questions is critical to choosing the best partner. Not all hunting providers are shaped alike. Cyber threat hunting is not simply deploying an endpoint solution and assuming it will solve all your problems. Every cyber threat hunting provider should have four primary capabilities that include: Deep direct experience with advanced adversaries and varied tactics, sweeping visibility into threatened environments, access to ongoing, active research driven by fled engagements, the ability to correlate data from many vantage points and cohesively analyze it. This paper describes the two model overlaps and both of them can meet dissimilar requirements that are well clarified in the following two subsequent subsections. The Cyber Threat Intelligence model is the foundation of the assessment course which is accomplished in the paper. [12] This white paper describes how Wilhoit's mindset, skills and approach to threat hunting research are starting to find their way into mainstream Security Operations Centers (SOCs), while assessing the long-term significance of this development for enterprise security and beyond. "Hunting can also help identify low-level vulnerabilities and high-level architectural issues, since hunters are finding ways attackers attempt to and successfully compromise the organization." [13] it has been examined that how a security establishment can improve their SIEM with a cyber-security platform in order to take control of the chaos, gain a completer empathetic of threats, remove incorrect positives, and form a proactive, intelligence-driven defense. Most enterprises use their SIEMs to collect log data and correlate security events across multiple systems (intrusion detection devices, firewalls, etc.), their internal security logs, and event data, and, as such, SIEMs provide a number of benefits. This whitepaper pursues in order to help the organization to decide whether their organization is ready to incorporate TH into their security or not [09]. A novel approach has been introduced in order to find out the executable pages which are up most important to any of the investigator [1]. In order to capture packets on the vitms's machine a software named wireshark has been used during the process of attack. After that the network capture has been saved as a pcap file. After that using that file, the communication in between the machines and the victim has been performed as an inspection in order to confirm whether the invader can be

detected or not as well as some other signs of the malicious activity can be found or not [5].

Table 1: Limitation of Existing Work

R.no.	Identification	Recommendation	Limitation
[1]	Give us complete overview of hiding techniques which prevents executable pages as well as complete examination of page table entries.	No recommendations	No proactive threat hunting Work on post Threats,
[2]	Memory forensics in the context of designing a forensic approach which will help to detect such advance malware threats, analyzation of sample memory image infected by a malware	Stealthy volatile attacks which many a time reside only in memory or exclusively run from the machine memory.	Proactive threat intelligence is absent, no ram processes.
[3]	Give us a complete overview of TC and also the extensibility of the platform to enable users to adapt and create automation for their processes, rather than forcing them to adapt their processes to ThreatConnect's paradigm.	Threat connect Tool	No RAM involved for hunting
[4]	Threat from insiders is a problem which is not going away, and all of the implications show that it is getting worse day by day. An effective insider threat justification approach will help diminish risk, as well as financial, lawful and in addition the potential destruction to the association	Automation using Kill 1 chain Process	No memory analysis
[5]	Hunt for a specific virus, Gh0st RAT. Forensic analysis of the memory using the Volatility software.	Using the Wireshark software.	Limited or no main memory analysis
[6]	Threat Hunting Survey shows that, for numerous organizations, it is still fresh and poorly defends from a course and organizational perspective. Unluckily, there are still numerous organizations that are still reacting to incidents damage instead of proactively looking for those threats inside the system. Threat hunting cannot be completely automated.	Statistical analysis of threat intelligence platforms. No recommendations	No proactive threat hunting
[7]	Business context is almost organization knowledge and its technical context is the footprint of malicious activities inside the organization network. Most Threat feed consist of IOC.	Different CTI tool like SEIM, slunk, toolkit, scanner, JSDidier, volatility and Wireshark etc. for network	Limited or no main memory analysis of business system
[8]	Memory image analysis. RAM acquisition tools like Memory Reader Belkasoft are used.	No recommendation	Root cause of attacks are missing
[9]	It emphasizes on methods for mixing and acting upon TI—information that should be vital to organizations of all sizes. The main key points of this paper are Defining TI, sourcing TI and making TI actionable.	Use “kill-chain process” like algorithms	Work on post threats

[10]	Conducted digital forensics, assisting investigators to identify crime scenes.	Four phases fusion framework, processes in digital forensics.	How these phases works are absent. No proper implementation is covered.
[11]	Analization of complex malwares by integrating static and memory forensics techniques.	Proposal of efficient and robust framework for the analization process	Information of malware types is absent.
[12]	The two model overlaps and both of them can meet dissimilar requirements that are well clarified in the following two subsequent subsections. The Cyber Threat Intelligence model is the foundation of the assessment course which is accomplished in the paper	No recommendation	Absent proactive threat intelligence
[13]	Wilhoit's mindset, skills and approach to threat hunting research are starting to find their way into mainstream Security Operations Centers (SOCs), while assessing the long-term significance of this development for enterprise security and beyond.	Domain Tools' Iris and wilhoit	Memory only Malwares left Vulnerable to fail proactive hunting
[14]	Using analytics and machine learning, specialists generate and work from baselines that help recognize new and abnormal patterns for faster detection and remediation of known and unknown threats.	Logs collection using security tools	Ram processes are not included

3 IMPLEMENTATION

The implementation phase has been divided in to two parts i.e. case 1 and case 2. In the first case a malware has been injected in to system in order to infect the system which won't be detected by the system main memory or the antivirus in the system in the VMware. In the second case the system stack has been overloaded by using the buffer flow technique and the host system will be on VMware and for this purpose Linux has been used and for the compilation python is used.

3.1 Case 1:

3.1.1 Injection of Malware

A malware is been injected in to the system by running it in to a hidden software by simply clicking on it as shown in the below figure which is a simple .exe file that contains malicious code which is been ran after opening this file and start its own process in to the system main memory. The aim of running this file is that it run in to system memory without leaving any artifacts for the existing security solutions and measures and it bypass all of the security measure of the operating system and it cannot be detected which in clearly a flaw in our existing systems as shown in the figure in the next column:

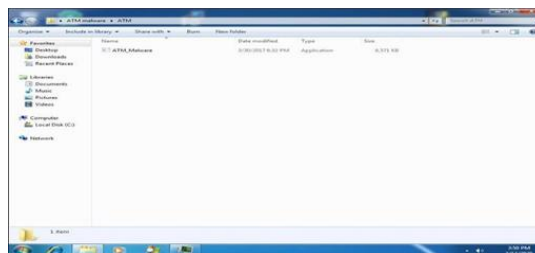


Fig. 1. Malware Injection

3.1.2 Acquiring memory dumps

In this phase the memory dumps have been taken in an external drive in order to do the prosed work. All of the memory dumps have been taken in an external drive in order to not interrupt the current process in the existing system. For the purpose of the collection of the memory dumps, red line tool has been used. After taking the memory dumps the next process is to acquire the memory dumps by using red line tool (a tool used for forensic analysis) which has been used to show all of the current process of the infected system main memory for the purpose of verification of the existence of the malware injected by our host system.

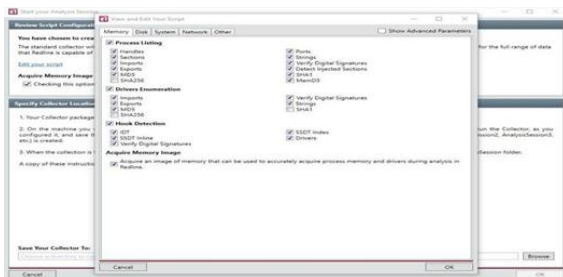


Fig. 2. Acquiring memory dumps

For this part of the research the dumps/file of the infected machine which has been saved in the external drive for the purpose of verification of the malicious activity in the infected system which hasn't been detected by the infected system main memory or the antivirus in the system. The investigation of the data has been performed by the red line tool. The investigation is a host based on an external investigation lead.

3.1.3 System information of the infected machine

After the second phase of the research which is investigation of the infected system. After that phase all of the information of the infected machine has been shown in the fig 3.4 which contains the machine information, operating system information, the user information and last but not least the bios information of the system. In the processes on the analyzation of the data a total of 2GB of the ram has been taken in order to save the processing time as the greater the ram size will be the more time it will take in the process of taking the memory dumps and the investigation of those dumps. Furthermore the operating system is also shown in the blow figure that which OS has been used in the infected system as well as the version of the bios. All of this information has been shown in the fig 3.4 as follow:



Fig. 3. System Information

Hence as it is already stated that this figure contains all of the information of the infected system that contains the system information etc. all of the information has been shown in the figure. Furthermore, it has been categorized as application process as well as services programs.

3.1.4 Verification of malware in the system

The final stage of case 1 is to verify the existence of the malicious file in to the system main memory. In the early phase a malicious code has been injected in to the system by using an exe file by simply clicking on it and all of the code inside that file has been ran in to the system main memory without leaving any artifact in the system which is a real threat to our system. Now in this phase it has been verified and shown in the figure that the file was running smoothly in to the system memory and our system was unable to detect it and point it as a malicious activity. Neither it has been caught by the antivirus as an abnormal activity. Now we can search the malware which is in a running process in the system memory with the same name as the system file "atm-malware" it merged itself with the other system files just as it is a normal activity in the running process of the system as shown in the figure. Without applying the technique of the memory forensics no other security tool or antivirus can detect it and prove it as a malicious activity. The below figure proves its existence in the current system.

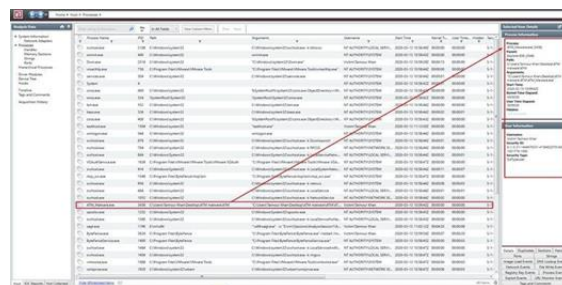


Fig. 4. Malware Verification

As shown in the above figure it has been verified that the file already was in a running process in the system and the system main memory failed to detect it as an abnormal activity which is a real flaw to our system. As it leaves a backdoor for the adversaries which can give a real harm to our systems. By using such files or code they can damage our system. All of the information of the file and the running process time has also been shown in the figure and it proves how our system are on risk for such attacks. The next step of this research contains the practical demonstration of getting access to a system by using code/ c program using Linux.

3.2 Case 2:

A system's stack will be overloaded by adding malicious code to run within authorized application

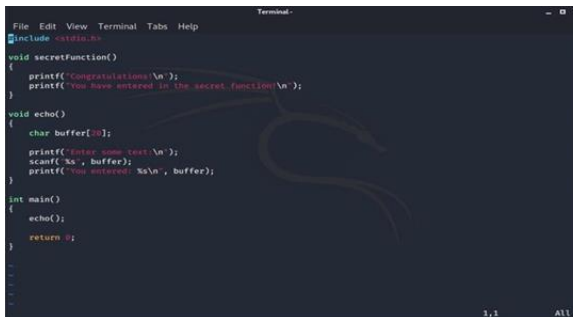
in the victim's RAM. When the code runs successfully it will make backdoor for the attacker to gain access to the target system without leaving any log information.

It refers to 'smash the stack' attack. On various C executions there is a possibility of the corruption of execution stack in some cases if you writing past the end of an array affirmed auto in an unchanging manner. The functionality of this code is set up to smashing the stack, and can cause return from the routine to jump to an unidentified address. This can create a few of the most insidious data-dependent bugs known to mankind. By doing this, attacker can cause program to run conferring to his will instead of the legitimate execution. Attacker may also inject evil code in to the program to perform their desired evil actions.

3.2.1 Practical Demonstration

Step 1

A file has been created using Linux in the first step of this practical which is named as vuln.c. the file must be a .c formatted file in order to open in it in c program as show in the below figure. All of the lines of the above code in the algorithm is written in to this file and later on ran in the terminal of Linux for the execution of this practical program. in this step the creation of the file has been done. Which can be seen in the fig 3.7 below:



```

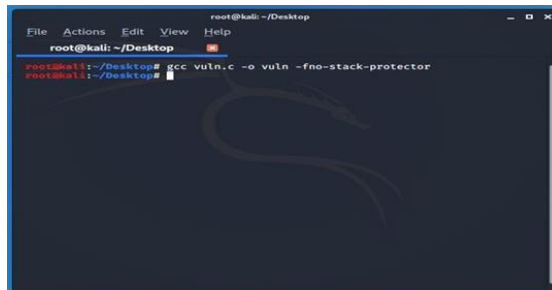
File Edit View Terminal Tabs Help
root@kali:~/Desktop
#include <stdio.h>
void secretFunction()
{
    printf("Congratulations \n");
    printf("You have entered in the secret function \n");
}
void echo()
{
    char buffer[50];
    printf("Enter some text \n");
    scanf("%s", buffer);
    printf("You entered: %s\n", buffer);
}
int main()
{
    echo();
    return 0;
}

```

Fig. 5. Creation of .c file

Step 2

In this step the compilation of the written program has been executed. By using this line of code i.e. gcc vuln.c-o vuln -fno-stack-protector. As shown in the below figure it has been executed successfully. In order to execute this code, it needs gcc which should be already installed in to the host machine.



```

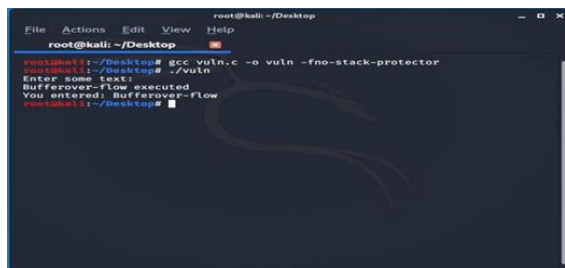
root@kali:~/Desktop
root@kali:~/Desktop
root@kali:~/Desktop# gcc vuln.c -o vuln -fno-stack-protector
root@kali:~/Desktop#

```

Fig. 6. Compilation of file

Step 3

Now as the creation of .c file and the compilation has already been done in the above two steps now it's time to execute the program. So, for a normal execution a very simple line of code has been used in order to smash the stack. Once it is happened the next step is all about the disassembly of the binary which has already been mentioned in the above algorithm part and the execution of my secret function after the stack is overflowed. The execution of the program after creation and the compilation is given in the below figure:



```

root@kali:~/Desktop
root@kali:~/Desktop# ./vuln
Enter some text:
Bufferoverflow-flow executed
You entered: Bufferoverflow-flow
root@kali:~/Desktop#

```

Fig. 7. Execution of program

Step 4

After the successful execution of the program it's time for the disassembly of the binaries. The disassembly of the binary is very necessary in such process in order to know the exact location of the secret function which has been written by us. the location of that secret function which we designed to call after the buffer overflow or smashing the stack. The code for this is objdump -d sts . the below is the figure of the disassembly of the binary which is clearly shown in the figure.

5 REFERENCES

- [1] Frank Block, Andreas Dewald “Windows Memory Forensics: Detecting (Un)Intentionally Hidden Injected Code by Examining Page Table Entries”, ELSEVIER, DFRWS 2019 USA.
- [2] Priya B Gadgil, Sangeeta Nagpure “Hunting advanced volatile threats using memory forensics”International journal of advance research , ideas and innovations in technology (Volume 4, Issue 4).
- [3] By Tony palmer, senior validation analyst; Alex Arcilla, Validation Analyst; and Domenic Amato, Associate Validation Analyst. February 2018, Threat Connect.
- [4] SANSWhitepaper,WrittenbyEricCole,PhD.
- [5] Jonathan Graham, Cheryl Hinds, Sandi Samuel” Hunting Malware: An example using Gh0st”International Conference on Computational Science and Computational Intelligence 2017.
- [6] The Hunter Strikes Back: A SANS Survey Written by Rob Lee and Robert M. Lee April 2017.
- [7] The Importance Of Business Information in Cyber Threat Intelligence University of Innsbruck, Department of Computer Science, Innsbruck, Austria.
- [8] Mr. Vivek Ravindra Sali, Mrs. H.K.Khanuja “RAM Forensics: The Analysis and Extraction of Malicious processes from memory Image using GUI based Memory Forensic Toolkit “ 2018. Fourth International Conference on Computing Communication Control and Automation(ICCUBEA).
- [9] Threat intelligence: What it is, and how to use it effectively Bromiley, M. (2016).
- [10] Da-yu kao, Yi-ting chao, Fuching tsai, Chia- yang huang, “Digital Evidence Analytics Applied in Cybercrime Investigations” 2018 IEEE Conference on Applications, Information and Network Security (AINS).
- [11] Mr. Chathuranga Rathnayaka, Dr. Aruna Jamdagni “An Efficient Approach for Advanced Malware Analysis using Memory Forensic Technique” 2017 IEEE Trustcom/BigDataSE/ICCESS.
- [12] M avroeidis, V., & Bromander, S. (2017). Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence. Proceedings of the IEEE.
- [13] The rise of threat hunting and why it mattersIn early 2017, written by DomainTools’ senior security researcher Kyle Wilhoit.
- [14] Closing the Skills Gap with Analytics and Machine Learning by ahmad tantawy October 2017.
- [15] SANS Threat Hunting & IR Summit 2018.
- [16] Salameh, Jamal N. Bani, "A New Technique for Sub-Key Generation in Block Ciphers," World Applied Sciences Journal 19, no. 11 , pp. 1630-1639, 2012.
- [17] Viegas, E. K., Santin, A. O., & Oliveira, L. S. (2017). Toward a reliable anomaly-based intrusion detection in real-world environmmts. Computer Networks, 127, 200-216.
- [18] E. Borgia, The Internet of Things vision: Key features, applications and open issues, Computer Communications 54 (2014) 1–31.
- [19] Kävrestad, J., Guide to Digital Forensics: A Concise and Practical Introduction, Springer International Publishing, pp. 3-8, 2107.
- [20] Priya B Gadgil, Sangeeta Nagpure “Analysis Of Advanced Volatile Threats Using Memory Forensics” Mahatma Education Society’s Transactions and Journals” Conference Proceedings ISBN 978-93-82626- 27-5”.
- [21] Dolly Uppal, Vishakha Mehra , Vinod Verma Basic survey on Malware Analysis, Tools and Techniques” nternational Journal on Computational Sciences & Applications (IJCSA) Vol.4, No.1, February 2014.
- [22] HaddadPajouh, Hamed, et al. "A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting." Future Generation Computer Systems 85 (2018): 88-96.
- [23] Liu, Qiang, et al. "A survey on security threats and defensive techniques of machine learning: A data driven view." IEEE access 6 (2018): 12103-12117.
- [24] Teoh, T. T., et al. "Applying RNN and J48 Deep Learning in Android Cyber Security Space for Threat Analysis." 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE). IEEE, 2018.
- [25] Arel, Itamar. "The threat of a reward-driven adversarial artificial general intelligence." Singularity Hypotheses. Springer, Berlin, Heidelberg, 2012. 43-60.
- [26] Homayoun, Sajad, et al. "DRTHIS: Deep ransomware threat hunting and intelligence system at the fog layer." Future Generation Computer Systems 90 (2019): 94-104.
- [27] Darabian, Hamid, et al. "A multiview learning method for malware threat hunting: windows, IoT and android as case studies." World Wide Web (2020): 1-20.