ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

Design of Advanced Cyber Threat Analysis Framework for Memory Forensics

Dr. Hardik Gohel, Dr. Himanshu Upadhyay

Postdoctoral Associate, Applied Research Center, Florida International University, United States (U.S.)

Sr. Research Scientist, Applied Research Center, Florida International University, United States (U.S.)

ABSTRACT - Malware, especially root kit, disturb system security by transforming kernel data structures to accomplish a variety of spiteful objectives. There is a various types of initial research has done on to identify malware modified data structures such as system call tables as well as function pointers. As a part of memory forensic the existing research accomplished only limited root kits and control data structures. This limitations leads us to do research and design a tool for advanced analysis of malware detection in data structures by using machine learning. This research focuses on applied operations of memory forensics in Linux machine and advanced data analysis using machine learning.

KEYWORDS: Cyber threats, Memory Forensics, Malware Analysis, Cyber security

I. INTRODUCTION

In today's world, the dependency on computers is growing extensively. Government agencies and private companies are attempting to protect themselves from cyber attacks with digital defense techniques like encryption, firewalls and heuristic or signature scanning, etc. Meanwhile, the number of attacks that include infiltrating military data centers, targeting power grids and stealing trade secrets from both private and public organizations continues to increase. The detection, response and reporting of these kinds of intrusions as well as other incidents involving computer systems, are crucial for cyber security professionals[1].

As these attacks continue to expand and the sophistication of the adversaries grow, defenders must adapt in order to survive. If proof-of-damage is never written to secondary storage, there is no way to rely on disk forensics. On the other hand, memory has a high potential to carry malicious code from an infection, partially or completely, even though it's never written to secondary storage (e.g., a hard drive). This is because the malicious code is loaded into the memory to execute. The random access memory of the victimized system also contains the proof of the system resources allocated by malicious code [2] [3]. Just like that, if the data taken from the organization is encrypted across the network, to determine which sensitive files were stolen and that won't be recognized by traditional packet capture techniques. However, passwords and encrypted keys can often be recovered by memory forensics, or even the file's plain-text contents before they were encrypted, providing information to understand the scope of an attack [4].

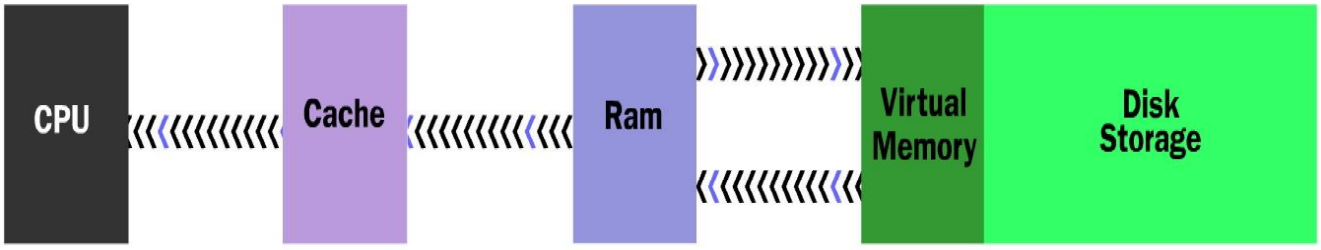

Figure 1: Flow of Memory Storage

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

II. MEMORY FORENSICS

In the world of digital forensics, memory forensics is arguably the most interesting and fruitful realm. Memory forensics involves analyzing the data stored in the physical memory at operating system runtime. Its primary application is in the investigation of advanced computer attacks which are stealthy enough to avoid leaving data on the computer hard drive. Consequently, the memory (RAM) must be analyzed for forensic information. Each and every function performed by an application or operating system results in a special kind of change to the random access memory. These changes often stay for a long time after completion of the operation, crucially storing them. Furthermore, memory forensics provides unprecedented visibility into the runtime state of the system, such as which processes were running, open network connections, and recently executed commands. Individuals can perform an extraction of these artifacts that is totally independent of the machine being investigated. It also reduces the chance of root kits or malware preventing the investigation process. Crucial data may exist exclusively in memory, such as unencrypted e-mail messages, disk encryption keys, non-cacheable internet history records, off-the-record chat messages and memory-resident injected code fragments [5].

Memory forensics is about capturing the profile as well as the memory contents and can add an invaluable resource to incident response, malware analysis, and digital forensics capabilities. Even though inspection of network packet captures and hard disks can yield compelling verification, it is often the contents of the computer memory that enables the full reconstruction of events, allowing an individual to determine what has already happened, what is presently happening, and what would happen with further infection through malware or an intrusion by advanced threat actors. For example, a piece of evidence found in RAM could help to associate typical forensic artifacts that may appear different and allow for an integration which could otherwise remain unnoticed [6].

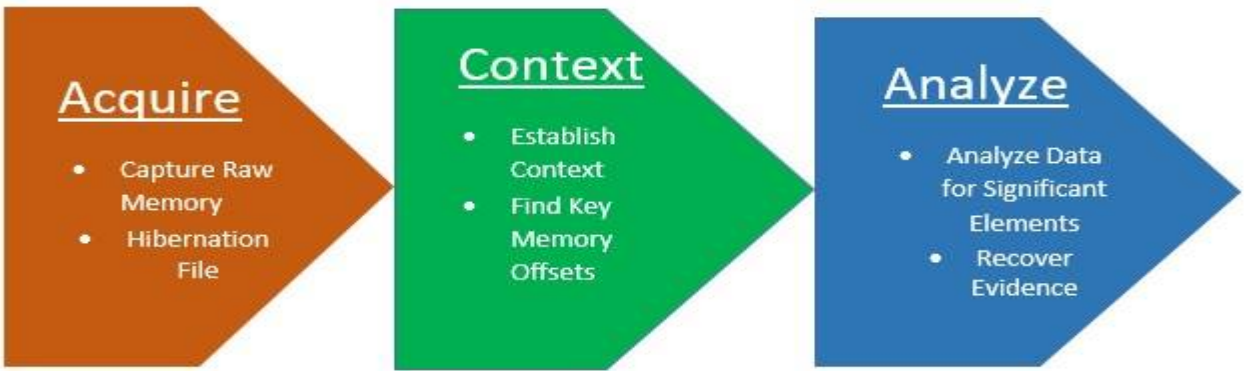

Figure 2: Process of Memory Forensics

There are three reasons for gathering and analyzing the data contained in the physical memory. First, the physical memory contains real-time data related to the operating system environment, such as the currently-mounted file system and the list of processes being operated. Second, even the encrypted data is generally decrypted when it is stored in the physical memory. Third, this method adapts well to the characteristics of embedded systems. Since an embedded system is rarely turned off, the data contained in the physical memory is mostly persistent. Therefore, significant information can be obtained if analysis is performed effectively on the physical memory [7].

The different types of information that can be extracted from memory include processes, dynamic link libraries (dll), process memory, image identification, kernel memory and objects, networking, registry, malware and root kits [8].

III. WHY MEMORY FORENSICS?

Everything in any type of operating system traverses random access memory, including processes and threads, root kits and malware, IP addresses, network sockets, URLs, open files, passwords, catches, clipboards and other user generated content, encrypted keys, configurations of hardware and software and windows registry keys and event logs.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

The types of artifacts found in memory dumps share a common origin. They all starts out as an allocation. Why, when and how the regions of memory were allocated sets them apart, in addition to the actual data stored within and around them. As a part of memory forensics, the study of these behaviors could be helpful to make inferences about the allocation of the content, leading to the ability to find and label specific types of data throughout a large memory dump. Additionally, the knowledge of allocation and de-allocation of memory and their algorithms (i.e., First Fit, Best Fit, Next Fit and Buddy System) could aid in understanding the context of the data. For example, which block of memory is free or which one is allocated [9] [10].

Figure 3 : Different OSs with Memory Forensics

Windows Forensics

Memory forensics for Windows involves finding and analyzing executive objects. Windows is written in C and the data as well as the attributes organization heavily use C structures. Out of those, several of structures are called executive objects. They are created, deleted and protected by the object manager of Windows. The object manager is a component of the kernel implemented by the NT module.

The major executive objects of Windows for memory forensics include file, process, symbolic link, token, window station, thread, desktop, mutant, types and keys. These executive objects are available with the corresponding name of the structure(e.g., file with _FILE_OBJECT structure). There is at least one Volatility plugin that analyzes each of the executive objects listed above [11].

Linux Forensics

In Linux memory forensics, the fundamental approach is to begin analyzing memory dumps of Linux. Specifically, one must be aware of traditional and modern memory acquisition techniques on Linux with their benefits and drawbacks. To perform Linux memory forensics, it is required to create Linux profiles, which are archives and contain useful information that Volatility needs to adequately find and intercept the data in memory dumps of Linux. Furthermore, one should be aware of the challenges to deploy Linux memory forensics in an enterprise environment, where critical servers may not even have compilers in C with other libraries which are found on workstations and desktops of standard Linux.

Mobile OS Memory Forensics

Various mobile operating system are currently in use, including Android, iOS, Windows 10 mobile, Tizen, Sailfish and Ubuntu touch. The most popular mobile OSs are Android and iOS. The rapid increase of systems running Android and Mac OS in both home and corporate environments has resulted in Android and Mac systems being a focus of targeted attacks. Because of these factors, cyber security experts have worked to develop tools for Android and iOS for robust investigative capabilities for Linux and Windows systems. To perform Android and Mac OS memory forensics, one has to create a Volatility profile for Android and Mac systems and can use one of the tools for acquisition of memory. Furthermore, some of the unique facets such as 64 bit addressing on 32 bit kernels, the typical user land and lay out of kernel address space, and the use of microkernel components are also major considerations.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

Tools of Memory Forensics

While there are multiple types of tools available to perform memory forensics, the Volatility framework and Linux Memory Extractor (LiME) are the most popular.

LiME is one of the best memory dump tools. It is a Linux kernel module (LKM) released by ShmooCon, which performs memory dumps for the Linux system. It is the first tool that can perform entire memory dumps from Linux-based devices and from Android. LiME is a powerful device that can perform memory dumps by loading modules immediately after compiling without any other operations, such as a change in kernel settings. Particularly in the case of Android, one can dump a file directly into external memory after inputting a pre-compiled module file into the external memory and loading the module through the command line. LiME features provide full memory acquisition and acquisition over the network interface with a minimal process footprint.

The second popular memory forensic tool is the Volatility framework. It is a single, cohesive framework that analyzes RAM dumps from Linux, 32- and 64-bit windows, Mac, and Android systems. The modular design of Volatility allows it to easily support new operating systems and architectures as they are released. So, all devices are targets. It doesn't limit the forensic capabilities to just Windows computers. Furthermore, it is an open source written in Python and has extensible and scriptable API with unparalleled feature sets and comprehensive coverage of file formats [12] [13].

Advanced Cyber Threat Analysis Tool

In a computer, data are stored in either its main memory unit or its auxiliary memory unit. RAM (Random Access Memory) is the main memory unit which retrieves the programs or data from the auxiliary memory unit and temporarily stores the information until the power is turned off. It is also called the physical memory. There are three reasons why gathering and analyzing the data contained in this physical memory is important during the step involving digital information collection. First, the physical memory contains data related to the real-time system operating environment, such as the currently-mounted file system and the list of processes being operated. Second, even the encrypted data are generally decrypted into plain statements when they are stored in the physical memory. Third, it conforms to the characteristics of embedded systems. Since an embedded system is rarely turned off, the data contained in the physical memory are not often volatilized. Therefore, significant information can be obtained if analysis thereof is performed effectively [14].

Memory forensics is forensic analysis of a computer's memory dump. Its primary application is investigation of advanced computer attacks which are stealthy enough to avoid leaving data on the computer's hard drive. Consequently, the memory (RAM) must be analyzed for forensic information.

The different types of information we can extract from memory are processes and dynamic link libraries (dll), process memory, image identification, kernel memory and objects, networking, registry, malware and root kits [15].

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

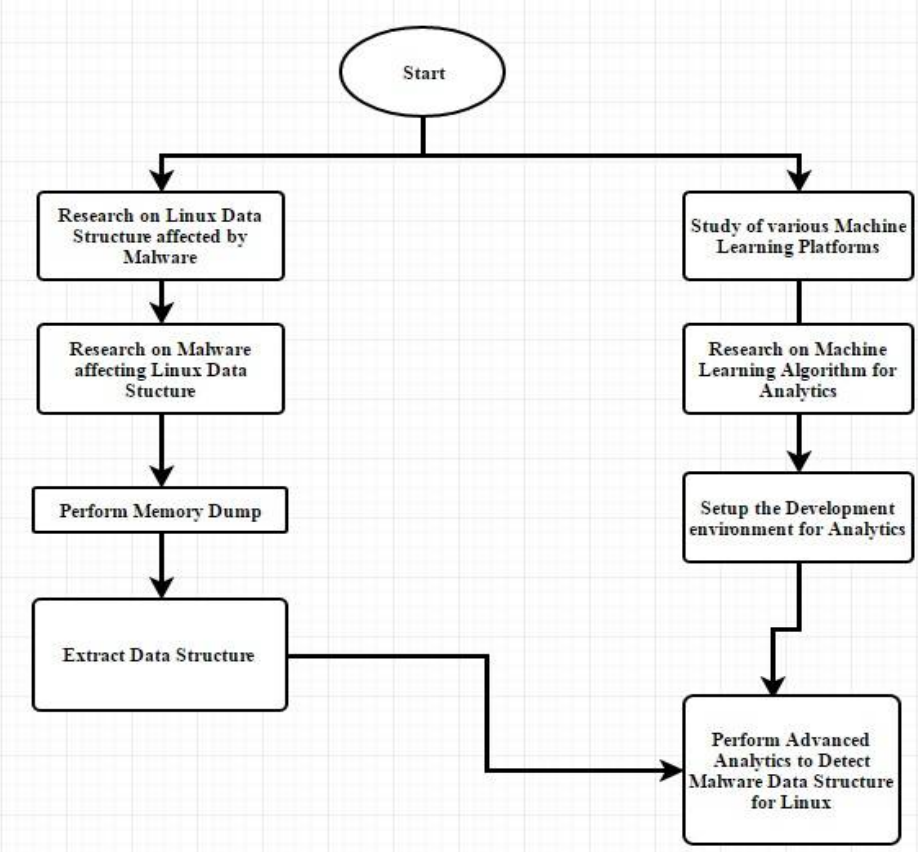

Figure 4 : Framework for Comprehensive tool execution

IV. RESULTS& OUTCOMES

Once the framework is narrowed down and tuned as a viable solution for malware detection, the results can be summarized to concisely describe the machine learning model as an answer to the malware detection. Research outcome in the data structure, algorithms that did or did not work or the model performance benefits achieved with data structure will be provided.

Once the model is found good enough at addressing the Malware detection, three key aspects to operationalize the model will be considered for developing the system. The three areas are the algorithm implementation, the automated testing of the model and the tracking of the model's performance.

The results - security tool, machine learning algorithm findings as well as data associated with the Linux kernel data structure can be shared with other Universities and students to help them pursue further research and enhancements of this system.

V. FUTURE RESEARCH& CONCLUSION

The major thrust area of operating system memory forensic is to perform Linux memory forensics. Government offices and business organizations are major Linux adopters. There are many security issues with the Linux operating system. The research direction should focus on applied operations of memory forensics in Linux machines and advanced data

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
An ISO 3297: 2007 Certified Organization Vol.5, Special Issue 2, April 2017
An International Conference on Recent Trends in IT Innovations - Tec'afe 2017
Organized by
Dept. of Computer Science, Garden City University, Bangalore-560049, India

analysis using machine learning which will be very useful to the Linux cyber society. This will allow government agencies, business organizations and also small-scale industries to secure their system operations. Antiviruses, which are one of the alternatives, can prevent malware from entering the system; however, what if an antivirus becomes disabled by admin access through the root kit? So, there is a need of research to provide models and algorithms to increase the security of the operating system resulting in the cyber protection of the users. Research focused on memory forensics using machine learning could be one of the key factor as it is rarely addressed by anti-virus products available in the market. In the cyber community, such research may provide the solution for many of the challenges.

REFERENCES

[1] Hardik Gohel. "Introduction to Network & Cyber Security", 2015
[2] M H Ligh, A Case, J Levy, A Walters. "The Art of Memory Forensics", 2014
[3] Mark Wade, "Memory Forensics: Where to Start" at http://www.forensicmag.com/ article/2011/06/ memory-forensics-where-start, 2011
[4] Gohel, Hardik. "Looking Back at the Evolution of the Internet." CSI Communications - Knowledge Digest for IT Community 38.6 (2014): 23-26
[5] Blackbag Team,"WINDOWS MEMORY FORENSICS", at https://www.blackbagtech.com/ blog/ 2016 /03/07/ windows-memory-forensics/2016
[6] Baliga, A., Ganapathy, V. and Iftode, L., 2011. Detecting kernel-level root kits using data structure invariants. IEEE Transactions on Dependable and Secure Computing, 8(5), pp.670-684.
[7] Hardik, Gohel. "Data Science - Data, Tools & Technologies." CSI Communications Knowledge Digest for IT Community 39.3(2015): 8-10
[8] Korkin, I. and Nesterov, I., 2015. Applying memory forensics to root kit detection. arXiv preprint arXiv:1506.04129.
[9] Hal Pomeranz, "Detecting Malware with Memory Forensics", at http://www.deer-run.com/~hal/ Detect_Malware_w_Memory_Forensics.pdf, 2015
[10] H Gohel, P Sharma. "Study of Quantum Computing with Significance of Machine Learning." CSI Communications - Knowledge Digest for IT Community 38.11 (2015): 21-23
[11] Joe Sylve https://github.com/504ensicsLabs/LiME
[12] Lime Forensics, https://code.google.com/p/lime-forensics/
[13] Volatility framework, https://github.com/volatilityfoundation/volatility/wiki
[14] Gohel H, Upadhyay H. "Cyber Threat Analysis with Memory Forensics" CSI Communication – Knowledge Digest for IT Community (2017).
[15] Hardik Gohel, Sanjay Bhatia, "Applied ICT – Beyond Oceans & Spaces", LAP Publications, 2017

BIOGRAPHY

Dr. Hardik Gohel is an Academician and United States based computer science researcher in the field of cyber security. Gohel has done investigative study of social media and also identified its security issues. Dr. Hardik has done more than 50 research publications and authored 3 books.