International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 4 (2015) pp. 9375-9382

http://www.ripublication.com

Detection Of Malware Based On Memory Mapping Technique

V.S.CHENCHU KARTHIK[1],

[1]Department of Computer Science
Sathyabama University
Chennai
INDIA
Email: chenchukarthik.me@gmail.com

A.PRAVIN[2]

[2]Faculty of computing
Sathyabama University
Chennai
INDIA
Email: pravin_ane@rediffmail.com

Abstract

Kernel malware is a major challenge to the systems as they process at the lowest most layers of the operating system. Some malwares can reuse the legitimate code and hence it confuse the behaviour of the malware hence it avoid detection. Kernel malware based on data centric was detected based on the data access patterns and its behaviour. Malware detect memory mapped technique provides an optimized solution to analyze kernel level code of the guest so and extract the malicious behaviour of the root kits. We propose a new technique which provides a slicing option to check the mapped memory by slicing step by step in kernel level.

***Key- words-*:** Kernel malware, Memory Mapper, Rootkits, Virtual machine monitor.

1.INRODUCTION:

Rootkits are the spiteful software that hides the program from detection. Malicious programs like viruses, worms have been used by the code injection attacks but some stealthy malware will reuse the legitimate code for the attack and hence avoid detection. kernel has the data structure that holds control data and jump tables are the target oriented by the rootkits[1]. A malware program has many options to make network based detection is very hard. The reason is the detectors cannot able monitor the activity of a malicious program directly[4]. There are many research proceeds on identifying malware activity like accessing data objects etc. but still there is no automated detection mechanism to eradicate the malware from the OS kernel. There is no technique to find the memory leakage and unauthorized changes in exe files in the system. Most advanced rootkits will hook the system call table of the OS and interrupt descriptor table and by hooking one interrupt, such that a clever rootkit can filter all kernel functions[7]. Irregular memory wastage is considered to be the some of the disadvantages. Which can be overcome by the Anti-Malware based on memory mappers which will frequently checks for memory management leaks, performance and unauthorized code execution.

2.RELATED WORK:

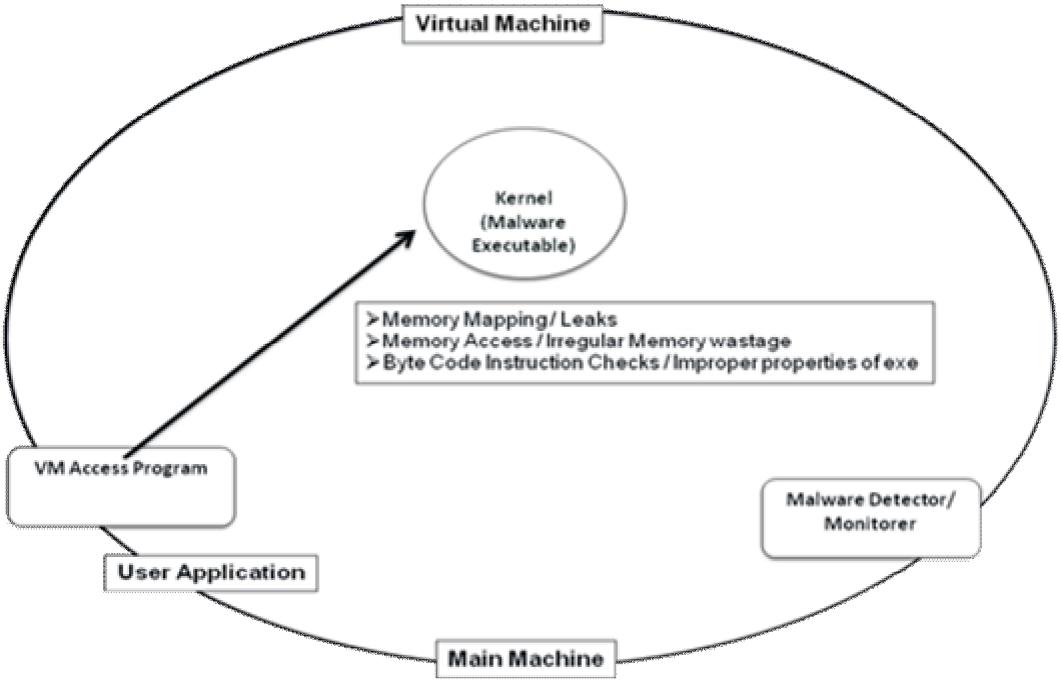
There are various approaches for detecting and analyzing the kernel malware . Junghwan Rhee et al proposed a system based on characterization, where it detects the malware based on the malware signature and detection of malware that are having similar data access patterns. Live Kernel object mapping is a technique which it montiors the kernel memory allocation by monitoring through HYPERVISOR. It examines the data behaviour by comparing both Benign kernel execution and kernel execution with malware such that it generates the data behaviour signature based on kernel execution this can be done by data behaviour aggregator and kernel memory mapper similarly in [6] it has a system called k-tracer that can dynamically analyze the kernel level code and extracts the malware. They proposed a technique that will slice and chop on specific execution paths in the kernel that will identifies all the datas that are infected by the malware. By comparing the above two techniques we got a plan to propose new Anti-Malware Detector that monitors application involves memory leaks, memory performance check, manipulated code execution. The Anti-Malware scanner is configured to check the virtual machine to detect the malware and affected files. It also slices the memory by both combination of backward and forward slicing to detect the malware efficiently. There is a tool that proposed in [1] called Gibraltor tool which is used for detection of rootkits in the kernel that modifies the data structure automatically. But it has the most disadvantage that time consumption is more to detect and prevent the malware. This drawback is overcome in the new

detector which we are proposing as we are monitoring the virtual computers from the main computer where it detects automatically. The automated system generating signatures for kernel level data structures, shows that the signatures modifying the structure contents will cause the OS to consider the object is not valid[2]. The gaps in coverage may leads to false signatures matching process. So based on the memory management check it will overcome the drawbacks like irregular memory wastage and it will increase the performance. The tool NICKLE which was discussed in [3] is a VM monitor based system that prevents the unauthorized kernel code execution in the guest OS. It has a new feature called memory shadowing, where it maintains a shadow physical memory for a running VM and performs the kernel level code authentication such that it is effective in preventing malware activity in guest system. The characteristics and the behaviour of the malware program to be build of an un-known program at runtime but it cannot be effective with some new malware [4]. So it can be easily modify the data structure of the kernel. The complexity of the memory mapped service in [5] comparing the runtime behaviour of a sample in the analysis system and it observes the interactions with the OS during runtime. Kernel data structure is the most common target which carried out by the kernel mode malware, so the system that analyze the kernel objects should consider the generic pointers and dynamic arrays etc.[12]. There are different types of attacks that have been discovered one of the is buffer overflow attack which will make the code not to respond. So we should adapt a new technique that virtually eliminates buffer overflow attack. However it cannot manage to restrict the stealthy malware which can penetrate easily in to the kernel level code and modifies the data structure of the kernel. Since the system got infected by the malware, we neither develop an application and perform various kinds of testing[10][12]. The various compliers such as GCC also affects by the some of the typical rootkits[11] will also be over-come by the malware scanner.

.

3.ARCHITECTURAL APPROACH:

The main machine application that will monitor the virtual machine program based on the memory performance check, memory management leaks, improper managed code such that it will sort down the issues in case of any problems when any suspicious thing happened in the VM



These are the list of modules that are going to be used for detection of malware from the main machine.

(a) Monitor creation module.

(b) Memory leak check module.

(c) Memory management mapping module.

(d) Invalid file properties validation module.

(e) Performance evaluation module.

A. Monitor Creation Module:

The proposed system algorithm called anti-malware scanner algorithm which will be configured to detect the virtual system's malware and affected files and eradicate it. The Live Kernel object Mapping System which can be used to monitor using a hypervisor which transparently and effectively get the memory related OS events and actions to generate a kernel object map. It dynamically monitors the kernel data objects of the guest OS all its allocation and de-allocations. There are some advanced kernel rootkits that hide themselves and cause damages to the memory of the virtual

machine. So we have consider this type of data hiding attack in to the consideration and monitor the guest OS effectively.

B. Memory Leak Check Module:

Memory leak check module can be used to detect the memory leakage which occurs due to the malware attack, it efficiently detect the malware content in the memory space. DKOM otherwise called as data hiding via direct kernel object manipulation which are hiding rootkits which is used to manipulate the kernel data structures. However, our scanner will effectively detects all the data hiding rootkit attacks. Dynamic detection of malware activity in virtual environment detects the vulnerable activity in kernel aided with proof carrying out over the injected malware code and memory leakage mechanism.

C. Invalid File Properties Validation Module:

Invalid file properties validation module will monitor the OS features related to memory management that varies due to malware injection. Once the attack initiated the properties of the kernel gets changed and it cannot perform efficiently.

D. Performance Evaluation Module:

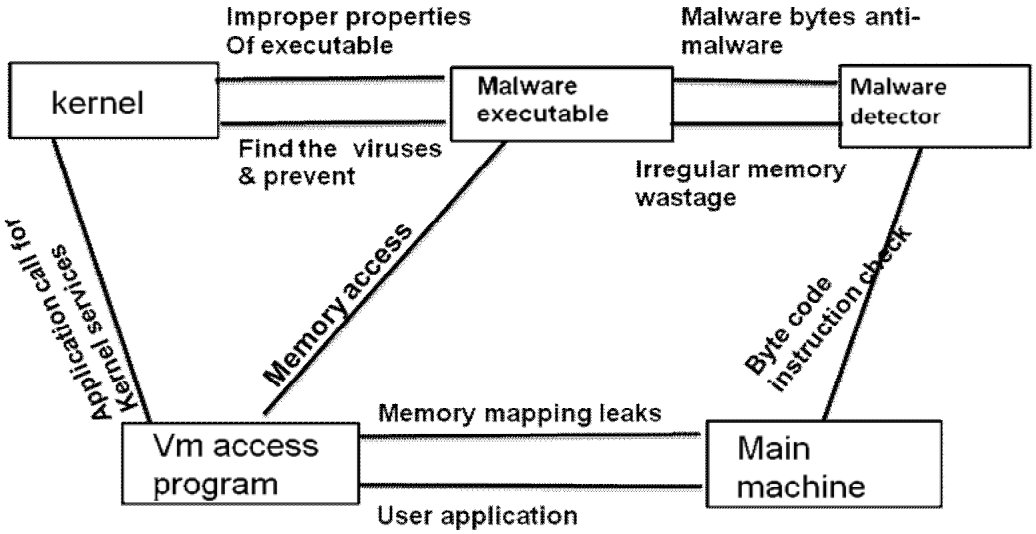
Performance can be measured using this module which will evaluate the operating system performance based on the runtime, efficiency, accuracy and reliability..

E. Memory Management Module:

In this module it will manage all the memory management task of the guest OS, it will monitor and manage the memory data structure of the virtual machine. It will be carried out the all possible slicing mechanisms in order to get the sensitive area which affect by the malware.

4.IMPLEMENTATION :

We have implemented the Anti-Malware scanner in our system which will effectively monitor and detect the malware using the memory mapping technique in the virtual machine. While our method is general and it is more enough to work with any OS that has a standard function call conventions e.g. Like Linux and Windows etc... Memory leaks leads to memory wastage in the virtual machine ad improper properties of execution file while accessing the VM access program are overcome by the malware anti-malware scanner using the memory mapping technique. There are various modules which we discussed earlier is going to be implemented, the overall structure.



In the above diagram, kernel will be executing malware detector which will find the improper executable files and it will check the byte code instruction check based on the memory mapping leaks.
INPUT:
P->Prevention program;
M->Malware;
OUTPUT: Prevent the malware using memory mapping technique.
Sliced = A;
If P(M) = 1then A //Sliced the affected memory files .

The system will be setup with anti-malware scanner that can dynamically analyze the memory management of the guest OS and extract the malicious behavior and also the sensitive area access and manipulation of code.

MALWARE DETECTOR:
The anti-malware scanner detects the execution of an unauthorized programs in the kernel and prevents it automatically. The Anti-Malware Scanner is configured to use the interpreted virtual memory states and the virtual disk states to detect the malware and the affected file. The instructions executed for the outside to the virtual machine. It will retrieve improper execution in the virtual machine. To solve this problem, we have developed a list of modules which we discussed earlier which will efficiently monitor and studies the signatures of the root-kits and its possible ways to attack the system. We can further simplify the detection program by we can call P as a perfect detection program and M is a malware , if P detects the malware then it will prevent it from the virtual machine automatically since it monitors the virtual machine from the main machine. There are some malware detectors which cannot able to detect accurately the malwares in the system, such that slicing up the memory and

comparing the original data structure and the one which got infected will fetch us the result. Once it able to pick up the slight variations then the malware detector will accurately find the source and the files got infected by the malware.

CONCULSION:

Malware detection using memory mapping technique is the most effective manner to detect and prevent the malware from the virtual machine. It consists of a new technique which will perform both backward and forward slicing option to check the mapped memory by slicing step by step in the kernel level. Such that it will identify the malware influenced sensitive data and possible solution to prevent the malware from the virtual machine. Thus the scanner which we going to implement in our system will effectively prevent the malware based on the memory mapping technique and the malware cannot be intruded by changing its data structure patterns and its access patterns.

REFERENCES:

[1] A. Baliga, V. Ganapathy, and L. Iftode, "Automatic inference andenforcement of kernel data structure invariants," in *Proc. 24th ACSAC*,Dec. 2008, pp. 77–86.

[2] B. Dolan-Gavitt, A. Srivastava, P. Traynor, and J. Giffin, "Robust signatures for kernel data structures," in *Proc. 16th ACM Conf. CCS*,2009, pp. 1–12

[3] R. Riley, X. Jiang, and D. Xu, "Guest-transparent prevention of kernelrootkits with VMM-based memory shadowing," in *Proc. 11th Int. Symp.RAID*, 2008, pp. 1–20.

[4] C. Kolbitsch, P. Milani Comparetti, C. Kruegel, E. Kirda, X. Zhou, andX. Wang, "Effective and efficient malware detection at the end host,"in *Proc. 18th USENIX Sec. Symp.*, Aug. 2009, pp. 351–366.

[5] D. Balzarotti, M. Cova, C. Karlberger, C. Kruegel, E. Kirda, andG. Vigna, "Efficient detection of split personalities in malware," in *Proc.17th Annu. NDSS*, Feb. 2010, pp. 1–17

[6] Junghwan Rhee, Ryan Riley, Zhiqiang Lin,Xuxian Jiang, and Dongyan Xu, "Data-Centric OS Kernel Malware characterization in *IEEE Transactions on Information Forensics and Security,vol.9,no.1,January 2014*

[7] M. Abadi, M. Budiu, Ú. Erlingsson, and J. Ligatti, "Control-flowintegrity: Principles, implementations, and applications," in *Proc. 12thACM Conf. CCS*, Nov. 2005, pp. 1–4.

[8] M. Carbone, W. Cui, L. Lu,W. Lee, M. Peinado, and X. Jiang, "Mappingkernel objects to enable systematic integrity checking," in *Proc. 16thACM Conf. CCS*, Nov. 2009, pp. 555–565.

[9] C. Cowan, C. Pu, D. Maier, J. Walpole, P. Bakke, S. Beattie,*et al.*, "StackGuard: Automatic adaptive detection and prevention ofbuffer-overflow attacks," in *Proc. 7th USENIX Sec. Conf.*, Jan. 1998,pp. 63–78.

[10] Albert Pravin, S. Srinivasan, "Effective test case selection and prioritization in regression testing," in *journal of computer science*.,may. 2013, pp. 654-659**.**

[11] J.Andrews, T. Sasikala, "Efficient framework architecture for improved tuning time and normalized tuning time ,"*WSEAS TRANSACTIONS ON INFORMATION SCIENCE*., Issue 7, Volume 10, July 2013.

[12] T. Prem Jacob, T. Ravi. Optimal Regression Test Case Prioritization using genetic algorithm.Life Sci Journal 2013;10(3),pp. 1021-1033 (ISSN:1097-8135).