Hypervisor and Virtual Machine Dependent Intrusion Detection and Prevention System for Virtualized Cloud Environment

[1]Ajay Kumara.M.A, [2]Jaidhar.C.D,
[1,2] Department of Information Technology, National Institute of Technology Karnataka
Surathkal, Mangalore, India
[1]ajayit13f01@nitk.edu.in, [2]jaidharcd@nitk.ac.in

Abstract*—Cloud Computing enabled by virtualization technology exhibits revolutionary change in IT Infrastructure. Hypervisor is a pillar of virtualization and it allows sharing of resources to virtual machines. Vulnerabilities present in virtual machine leveraged by an attacker to launch the advanced persistent attacks such as stealthy rootkit, Trojan, Denial of Service (DoS) and Distributed Denial of Service (DDoS) attack etc. Virtual Machines are prime target for malignant cloud user or an attacker to launch attacks as they are easily available for rent from Cloud Service Provider (CSP). Attacks on virtual machine can disrupt the normal operation of cloud infrastructure. In order to secure the virtual environment, defence mechanism is highly imperative at each virtual machine to identify the attacks occurring at virtual machine in timely manner. This work proposes In-and-Out-of-the-Box Virtual Machine and Hypervisor based Intrusion Detection and Prevention System for virtualized environment to ensure robust state of the virtual machine by detecting followed by eradicating rootkits as well as other attacks. We conducted experiments using popular open source Host based Intrusion Detection System (HIDS) called Open Source SECurity Event Correlator (OSSEC). Both Linux and windows based rootkits, DoS attack, Files integrity verification test are conducted and they are successfully detected by OSSEC.

Keywords— ***Cloud Computing; DoS Attack; Hypervisor;* Intrusion *Detection and Prevention System; Rootkit; Virtual Machine; Virtualization;***

I. INTRODUCTION

Cloud computing improves resource utilization while reducing infrastructural cost. It is based on existing technologies such as service-oriented architecture, virtualization and utility computing. Virtualization is the key underlying technology for cloud computing architecture. It facilitates sharing of physical machine resources such as CPU, Memory, I/O and Network interface etc., among several virtual machines running on the same physical machine. Protecting virtualized resources of Guest Operating System (GOS) against advanced sophisticated attacks is a massive challenge for Cloud Service Provider (CSP).

Hypervisor is a pillar for virtualization [1] and allows sharing of resources to virtual machines. In addition, it offers greater security risk to CSP. Once hypervisor is compromised from miscreant (intruder), the entire virtual environment is under the control of an attacker. In cloud computing environment, intrusion may originate from multiple sources such as virtual machine [2], virtual network [3], malicious hypervisor [4], etc. Virtual machines are prime target for an attackers as they are easily available for rent and VMs are directly accessible to external world through CSP [5]. Attacker adopts various tricks to uncover virtual machine vulnerabilities so that exploited vulnerabilities can use as door to compromise the virtual machine. Once virtual machine is under the control of an attacker, he/she can inject the malware (rootkit or spyware or virus or worms or Trojans etc.) to disorder the normal operation of the virtual machine. Further, attacker can trespass virtualized environment through compromised virtual machine. Only anti-malware (malware detection software) is scanty to maintain healthy state of virtual machine all the time. To protect the entire virtual machine against various types of threats and attacks, Intrusion Detection and Prevention System (IDPS) is prime requirement [6]. The IDPS should have the capability of malware detection (rootkit detection, spyware detection, virus, worms, Trojan etc.), log analysis, file integrity checking, incoming traffic analysis, active response etc. [16]. In addition, it should have detection capacity of unknown attacks as well as known attacks. Only detection is inadequate to safeguard the virtual environment without prevention. In this work, In-and-Out-of–the-Box virtual machine based intrusion detection and prevention approach is proposed for virtualized environment. Management unit of hypervisor (Out-of-the-Box) automatically deploys Intrusion Detection and Prevention System (IDPS) onto newly the created virtual machine (In-the-Box) as and when they launched.

The main objective of the virtual machine IDPS is to identify and take away any kind of intrusions in real time so that healthy state of the virtual machine is maintained continuously. For instance, stealthy rootkits, DoS attack, File integrity violation etc. should be removed immediately as and when detected. This intern safeguard the hypervisor by detecting and eradicating intrusion at virtual machine level. We have conducted the experiments for Linux and windows rootkits including (DoS) attack and file integrity verification. OSSEC [16] has used as virtual machine based Intrusion Detection System (IDS) and it has successfully detected the injected intrusions. Moreover, it recorded the identified intrusion details onto log files. The rest of the paper is organized as follows. Section II provides background for

978-1-4799-7315-6/15/$31.00 ©2015 IEEE

hypervisor, rootkits and intrusion detection and prevention system. Section III illustrates previous work related to rootkit detection methods. Section IV provides our proposed In-and-Out-of-the-Box virtual machine and hypervisor based intrusion detection and prevention system for virtualized environment. Experimental setup and results are discussed in section V. Discussion made in this work explained in section VI. Finally, section VII concludes the paper with future work.

II. BACKGROUND

A. Hypervisor

Virtual machines running on the same physical machine are under the control of software called hypervisor or Virtual Machine Monitor (VMM). The Hypervisor is a software component used to ensure sharing of host system resources in a virtualized environment among virtual machines [7]. Major tasks of the hypervisor are virtual machine creation, termination, migration and resource allocations etc. The hypervisor is classified into Type-I hypervisor and Type-II hypervisor based on the place of operation [5].Type-I hypervisor runs directly on the bare hardware of the physical machine (host system) as base operating system. It directly interacts with physical hardware for scheduling and allocating resources for any virtual machine. Type-II hypervisor is a special type of software operates above Host Operating System (HOS).

B. Rootkits

Rootkits are one type of malware that runs at highest privileges access of operating system (kernel or ring '0') by exploiting the security weakness present on the system. To maintain persistent and undetectable presence on the victim system. It hides the existence of certain processes or programs or files or ports or directories etc. Rootkits deviates normal behavior of the system by injecting malicious code into an operating system or application software. Some of the rootkits hides in anti-malware to evade the security check [8]. Rootkits are mainly classified into user-mode rootkits (application-level) and Kernel-level rootkits [9].

User-mode rootkits modifies or intercepts operating system critical files to hide themselves since they unable to modify kernel structure directly. For instance executable files or system libraries (DLL file on windows operating system). Kernel-level rootkits alters the code of core operating system i.e. kernel device driver, system call table, kernel code etc. Severity of Kernel-level rootkits are increasingly high as compared to user-mode rootkits as they modifies operating system source code and they are difficult to detect due to their hidden existence. Detection and prevention of advanced rootkits is the greatest challenge for security vendor. From fast few years number of rootkit detection and prevention techniques are proposed [9][12]. Rootkit detection methods in virtualized environment are broadly classified into In-the-Box-rootkit detection and Out-of–the-Box rootkit detection [9]. The In-the Box detection approach is able to scan and identify the illicit activities occurring within in the system without external entity support. Out-of-the-Box rootkit detection method make use of external agent to monitor virtual machine by means of Virtual Machine Introspection (VMI) or from outside the virtual machine. Traditional protective measures are ineffective to detect and confiscate advanced rootkits. Variant of Agobot rootkit [10] is able to spot and eradicate more than 100 types of anti-malware. Since decade researchers have proposed various approach for rootkit detection in virtual machine [12][13].

C. Intrusion Detection and Prevention System

Intrusion can be defined as unauthorized access to comprise Confidentiality, Integrity and Availability (CIA) or disrupt the normal operation or bypass the security measures currently being running on host system/networking device [6]. The function of the IDPS is to detect and prevent intrusion in a timely manner to avert the damage. Intrusion may originate from an attacker or it may be self-propagating (Ex. Virus or Worms). Two main types of intrusion detection techniques are signature based and anomaly based [14].

Signature based approach detects intrusion by comparing the observed signature with pre-defined signatures present in the database. It detects known attacks effectively. However, it cannot detect new attacks without the corresponding signature in the signature database. Continuous updation of signature database is crucial to detect new attacks. This limitation is overcome by anomaly based [15] intrusion detection system. It detects intrusion by comparing observed activities with baseline profile without signature database. The outcome of the comparison is anomalous when observed activities exceed the threshold otherwise legitimate. The major advantage of anomaly based intrusion detection system is that it can detect new attacks including zero day attack [15].

III. PREVIOUS WORK

Xueyang et.al. have proposed Virtual Machine Monitor (VMM) based framework with a name NumChecker [17]. It detects Kernel-level rootkits by observing a number of certain hardware events occur during the execution of system call. For instance, total number of instructions, floating point operations, number of branch operations as well as type of branch operation, returns etc., are monitoring hardware events. The deviation of monitored hardware events count during execution of a system call from normal count indicates the presence of malicious events. Similarly M.Schmidt et.al [12] have proposed malware detection and kernel-level rootkit prevention approach for virtualized cloud computing environment.

C. Kruegel et.al. [18] have proposed Kernel-level rootkit detection approach that relies on statistical analysis of Loadable Kernel Module binaries. It ascertains malicious kernel module before they enter into kernel space without accessing module source code. It utilizes behavioural specification and symbolic execution that provides additional protection against malicious modification of the kernel. Musavi, S.A et.al. [13] have proposed static analysis based technique for kernel-level rootkit detection (malicious kernel driver). A. Baliga et. al [19] have developed a tool Gibraltar for detecting kernel-level rootkits based on data structure invariants technique. It uses anomaly based detection approach to detect both control and non-control invariants of kernel data structure. Major limitation is that Gilbart is unable to detect transient attacks and authors also elaborated other limitations in their paper [19].

978-1-4799-7315-6/15/$31.00 ©2015 IEEE

Garfinkel et.al. [20] have proposed a virtual machine introspection based architecture that provides isolation, inspection and interposition properties of virtual machine monitor. The introspection feature offers VMM based IDS to inspect the entire state of the virtual machine by functioning at hypervisor level. It is one of the promising approach to secure virtual machine from intruder. However this technique face the semantic gap of VMM data and guest VM software view [20].

IV. PROPOSED IN-AND-OUT-OF-THE–BOX VMIDPS

Every day the number of unknown threats and malware are keeps launching by an attacker to supersede the existing defence mechanisms. In order to resist new and old threats as well as attacks, intelligent IDPS is highly indispensable. In this work, Virtual Machine and Hypervisor based Intrusion Detection and Prevention System (VMIDPS) for virtualized environment is proposed and described in this section. Proposed architecture is as shown in Fig1.

The management unit is one of the components of the hypervisor and IDPS core resides on it. The VMIDPS-server is an integral part of the IDPS core and it runs on hypervisor. The hypervisor informs the management unit to deploy IDPS-agent onto the new virtual machine whenever new virtual machine is launched. IDPS running on virtual machine is named as Virtual Machine based Intrusion Detection and Prevention System (VMIDPS). The VMIDPS scans the entire virtual machine to certify that system is in robust as well as uninfected state. Virtual machine allows for function only if it is certified as robust system else VMIDPS triggers an alert to take appropriate action to bring back the virtual machine to normal state. VMIDPS on each virtual machine continuously monitors and analyses occurring events to detect and avert malicious events in real time. Multiple intrusion detection techniques such as file integrity verification, signature based intrusion detection and anomaly based intrusion detection are adopted in VMIDPS to detect various types of intrusion(Rootkits, Virus, Worms, Ports scan, File alteration and so on).The VMIDPS periodically sends the entire state of the virtual machine to VMIDPS-Server to detect intrusion that are bypassed at virtual machine level. VMIDPS-server utilize cross-view analysis based intrusion detection technique to spot the intrusions.

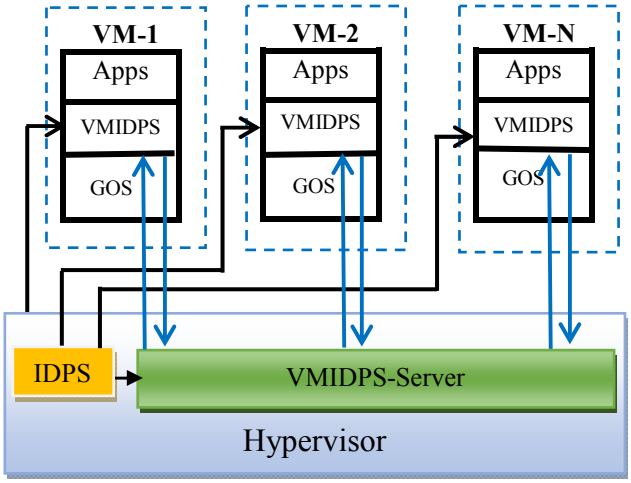

Fig .1. Architecture of VMIDPS for Virtualized Environment

A. File Integrity Verification

VMIDPS periodically monitors operating system files as well as other critical files to ensure their integrity. It is achieved by computing and comparing their cryptographic hash digest with pre-computed cryptographic hash digest. Comparison mismatch demonstrates file content alteration.VMIDPS triggers an alert as and when file integrity violation noticed. File integrity verification steps are shown in Algorithm-1. However, file integrity verification does not provide accurate information regarding type of intrusion (what content of the file modified). Therefore, files whose integrity is violated are further analyzed to uncover more details of the intrusion.

B. Signature based Intrusion Detection System

Signature based Intrusion Detection System (SIDS) is also known as misuse detection. It detects intrusion by comparing the observed signature with pre-defined signatures present in the database. Operation of SIDS is as shown in Algorithm-2. It detects any kind of known attacks effectively. However, it cannot detect new attacks without the corresponding signature in the signature database. Continuous updation of signature database is crucial to detect new attacks. This limitation is overcome by Anomaly based Intrusion Detection System.

C. Anomaly based Intrusion Detection System

It detects intrusion by comparing observed activities with baseline profile without signature database. 'Th_i' is a threshold for an activity **"i"** where **i**=1,2,3,4,5,…..N. Profile of the system **PS** = $\{Th_1, Th_2, Th_3 \ldots Th_N\}$. AS_i is an anomaly score of an activity 'i' observed for the period of time **'T'** is

$$As_i = \sum_{i=1}^{K} ac_i \quad K=1, 2, 3, 4, 5...,$$

Where" ac_i" is an observed activity 'i'.

$$f(As_i, Th_i) = \begin{cases} \textbf{Anomalous}, & As_i > Th_i \\ \textbf{Normal}, & As_i \leq Th_i \end{cases}$$

Where 'f ' is a comparison function. The outcome of the comparison is anomalous when observed activities exceed the threshold otherwise legitimate. Major advantage of anomaly based intrusion detection system is that it can detect new attacks including zero day attack.

Algorithm-1: File Integrity verification algorithm

Notations used in this algorithm are as follows

1: F_i → Integrity of the file to be checked

2: h( ) → One-way Hash function

3: hD' → Hash digest of file "F_i" stored in the hash digest database [h(F_i) → hD']

4: hD"→ Newly computed hash digest for the file "F_i" by one-way hash function h( ).

5: N → Number of files those integrity to be checked

S1: Input: File "F_i" integrity to be checked
Hash digest database for the file "F_i".

S2: For (i=1; i ≤ N; i++)
{

S3: Compute hD" and compare it with hD' (hD"= =hD')

S4: **If** the comparison is true goto S2
else
S5: Declare file "F_i" integrity is violated and display both hD" as well as hD'
S6: Send file "F_i" to signature based intrusion detection module to spot the malicious content present in it.
}
S7: Halt

Algorithm-2: Signature Based Intrusion Detection

The notations used in this algorithm are as follows
1: F_i → File to be checked
2: SD → Signature Database of F_i
3: M → Number of files to be checked
4: FSD_i → Signature of the file F_i

SS1: **Input:** File "F_i" to be checked and Signature database "SD"
SS2: For (i=1; i ≤ M; i++)
{
SS3: Extract FSD_i for the file "F_i" and compare it with SD.
SS4: if the comparison is match with one or more entries of the SD **then** declare the corresponding attack
}
SS5: Halt

D. Cross-View Analysis

Each virtual machine periodically sends its entire state information to VMIDPS-server. Once VMIDPS-server receive VM information then it request the hypervisor to supply actual low level information of the particular VM. As hypervisor has complete control on every VM, it supplies requested VM information to VMIDPS-server. By comparing information supplied by the hypervisor with information received from VM, it identifies the intrusion if the comparison result is deviated.

V. EXPERIMENTAL SETUP AND RESULTS

In this work, Oracle Virtual Box [22] was used to setup the virtualized environment and virtual machines were used to conduct the experiments. Host based Intrusion Detection System such as OSSEC [16] was used as Virtual Machine based Intrusion Detection System in this experimental work. OSSEC supports server, agent, local and hybrid type of installation. A local type of OSSEC installation was adopted to simulate the In-the-Box detection approach. In this method, each virtual machine has its own dedicated IDS which monitors the entire system events and triggers an alert as per the conditions set in the configuration file. Server type of installation was followed to simulate Out-of-the-Box intrusion detection. In this approach, OSSEC-agent is needed on each virtual machine. OSSEC-agent captures the events occurring on a virtual machine and sends them to an OSSEC-server to ascertain the malicious event trace. OSSEC-server and OSSEC-agent communicates securely using the key shared between them. Prevention is achieved by removing the detected intrusions.

In this experimental work, both Linux and Windows based Rootkits (Table.I), DoS attack, and File Integrity verification test were conducted on Linux and Windows guest operating system using oracle virtual box hypervisor 4.3.18 [21].

A. Linux rootkits

Kernal Beast (KBeast) Rootkit [22][11]: It is Kernel-level rootkit and has the ability to hide Loadable Kernel Module (LKM), Process (ps, pstree, top, lsof), Ports, Socket and connections (netstat, lsof) and Files/Directory. Anti-kill process, Anti-remove files, Anti-delete loadable kernel modules, local root escalation backdoor and remote binding backdoor hidden by the kernel rootkit are also features of KBeast rootkit. In addition, it has the capacity to launch password protected backdoor.

KBeast rootkit injected onto UBUNTU 10.04 32-bit operating system virtual machine. Fig. 2a gives the screenshot of KBeast Rootkit installation completion. Fig.2b provides the snapshot of *netstat* command output in that port number *"14477"(attacker defined)* is completely hidden even though the connection established with port number 14477. Kbeast rootkit file "*ipsec-kbeast- v1.c"* is unable to delete using remove command *'rm'*. OSSEC server successfully detected the injected rootkit and alert message also generated. Fig 2c. Shows the triggered alert messages in connection with detected rootkit.

B. Windows's rootkit

Hacker Defender [23] is a persistent user mode rootkit of windows system, it modifies the native Application Program Interface (API) function of windows system. Its major goal is to allow hacker to hide process, files, registry keys system driver. Moreover, it checks open port of the network connection This rootkit consists of two files one is executable files "hxdef100.exe" and another one is configuration files "hxdef100.ini" which is used to set the attributes for rootkit. This rootkit successfully injected onto Windows-7 operating system based guest virtual machine. The injected executable file creates new process that hides the process. Screenshot illustrated in Fig.4a provides the process details before and after the injection of the rootkit. Alert message generated by OSSEC after the rootkit detection is as shown in Fig.4b.

C. DOS Attack [24]

Intention is to make computer or network resource unavailable to legitimate users by flooding enormous bogus messages. Detecting and preventing DoS and DDoS attack is massive challenges for cloud service provider. Attackers may use sophisticated powerful DoS attack tool to flood massive number fake requests onto victim machine/server so that service disruption arises to legitimate one.

LOIC (Low Orbit Ion Cannon) [25] is one such open source tool to launch massive number of UDP, TCP and HTTP attack towards target machine. In order to test the feasi-bility of our proposed system, we have launched DoS attack. It is successfully initiated using LOIC from windows-7 operating

978-1-4799-7315-6/15/$31.00 ©2015 IEEE

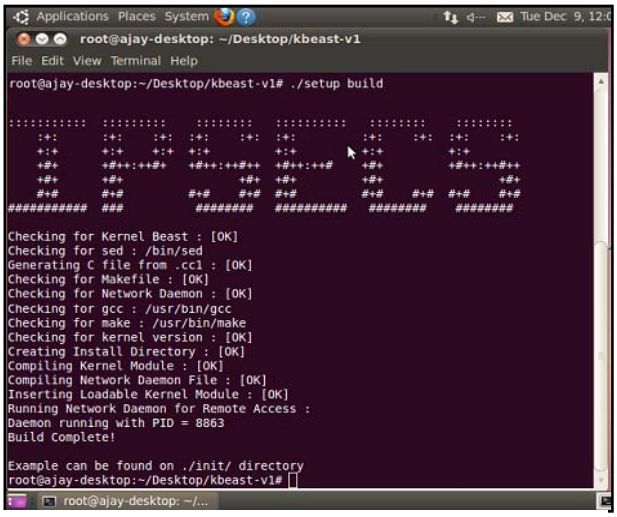

Fig. 2a. KBeast Rootkit Installation Completion

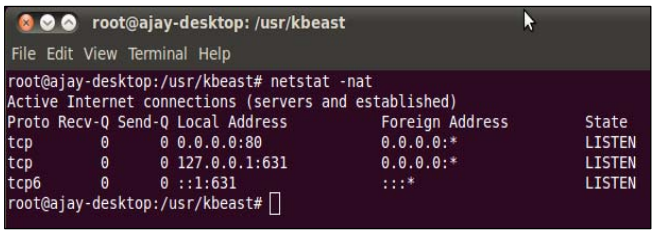

Fig.2b. Rootkit Hidden the Port number

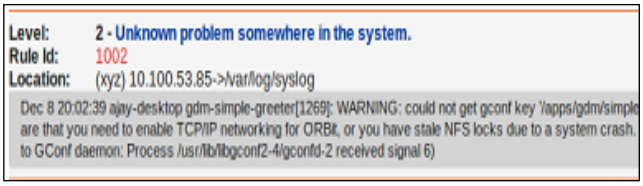

Fig. 2c. OSSEC Alert Message for Rootkit Injection

Fig. 2. KBeast Rootkit Experiment

system machine as a sending host towards target machine Ubuntu 12.04 LTS server desktop. Fig. 3 shows the notice triggered by OSSEC after successful detection of DoS attack.

Alert 1417094102.129619 : mail - apache, invalid_request, 2014 Dec 11 19:45:02 ituser-HP-Pavilion-dv6-Notebook-PC->/var/log/apache2/error.log Rule: 30117 (level 10) -> 'Invalid URI, file name too long.' Src IP: 10.100.54.7. [Thu Dec 11 19:45:01 2014] [error] [client 10.100.54.7] **request failed: URI too long (longer than 8190)

Fig .3. OSSEC detected DoS attack as URI too long

VI. DISCUSSION

The proposed Virtual Machine and Hypervisor dependent Intrusion Detection and Prevention System assumes that hypervisor (Oracle Virtual box 4.3.18) is trustworthy and is capable to send low level information such as CPU, Memory and Disk information to OSSES-Sever. The second assumption is that management unit of hypervisor automatically deploys OSSEC IDPS onto new virtual machines as and when they launched and OSSEC server runs on hypervisor.In our proposed model we have tested Linux and windows rootkit and DoS attack. Successful detection of these intrusions depicted in experimental section. We believe that proposed model also detect other types of intrusion and real world advanced persistent threats.

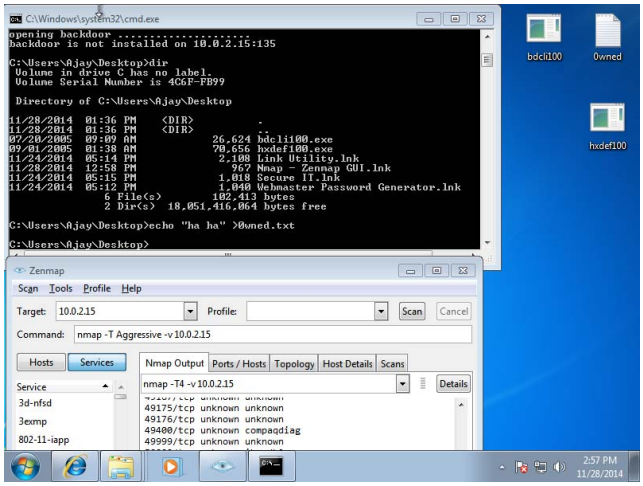

Fig .4a. Hacker Defender Rootkit Injection

** Alert 1418200710.26154: mail - windows, 2014 Dec 10 14:08:30 (Laptop) 10.100.55.32>WinEvtLog Rule: 18147 (level 5) -> 'Application Installed.' User: ITDEPT 2014 Dec 10 14:08:28 WinEvtLog: Application: INFORMATION(11707): MsiInstaller: ITDEPT: ITDEPT-HP: Installation completed successfully.

Fig .4b. OSSEC alert message for Hacker Defender Rootkit

Table I. Compression of Windows and Linux Rootkits Tested in this experiment

Rootkit Name	Virtual box 4.3.18	Guest VM		Agent running	Kernel Mode	Hides Process	File Hide	Hides File	Hides Port	Privilege Escalation	OSSEC detected ?
		Windows	Linux								
KBeast	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
Hacker Defender	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓

978-1-4799-7315-6/15/$31.00 ©2015 IEEE

VII. CONCLUSION AND FUTURE WORK

In this paper, In-and-Out-of-the-Box virtual machine and hypervisor dependent intrusion detection and prevention system is proposed. The main goal of this work is to detect real time intrusions including stealthy self-hiding rootkits and DoS attack. These kind of intrusions are potentially harmful for vitalization environment those are originating via virtual machine. The architecture of VMIDPS leverage cross view based technique for real time identification of intrusion. File alternation identification is achieved by integrity checking algorithm. The proposed approach ensure healthy state of virtual machine by detecting and eradicating intrusion in real time. OSSEC IDS effectively utilized as VM based IDS to identify the abnormal activities of VM. Experimental results shows the detection capability of VMIDPS.

The same work continues as a future work with Virtual Machine Introspection (VMI) technique as Out-of-the-Box intrusion detection technique. More number of experiments are planned to conduct for different attack scenario (hypervisor and VM based) to measure the efficiency of VMI based detection approach. Different malicious software such rootkits, malware, spywares and warm are planned to test in future work to verify the detection as well as prevention capability of intrusion.

REFRENCES

[1] http://www.zdnet.com/article/hypervisors-are-the-pillars-of-the-cloud-not-the-achilles-heel/

[2] Bahram, S., Jiang, X., Wang, Z., Grace, M., Li, J., Srinivasan, D., Rhee, J., Xu, D., "DKSM: Subverting Virtual Machine Introspection for Pun and Profit", In Proceedings of 29th IEEE Symposium on Reliable Distributed Systems, pages 82–91, IEEE Computer Society, Washington, DC, USA (2010). DOI:10.1109/SRDS.2010.39

[3] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "Nice: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems", IEEE Transactions on Dependable and Secure Computing (TDSC), Vol. 10, No.4, Pages 198-211, 2013.

[4] Ahmed M. Azab , Peng Ning , Zhi Wang , Xuxian Jiang , Xiaolan Zhang ,Nathan C. Skalsky, "HyperSentry: Enabling Stealthy in-context measurement of Hypervisor Integrity", Proceedings of the 17th ACM conference on Computer and Communications Security, October 04-08, 2010, Chicago, Illinois, USA [DOI:10.1145/1866307.1866313].

[5] Wesley Vollmar, Thomas Harris, Lowell Long,and Robert Green 2013, "Hypervisor Security in Cloud Computing Systems, ACM Computing Surveys", ACM Comput. Surv. 0, 0, Article 0 (2014), 22 pages.DOI:http://dx.doi.org/10.1145/0000000.0000000.

[6] Scarfone K, Mell P. "Guide to Intrusion Detection and Prevention Ssystems (IDPS)". NIST Special Publication 800-94; 2007. http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf

[7] http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

[8] Kim S, Park J, Lee K, You I, Yim K. "A Brief Survey on Rootkit Techniques in Malicious Codes", Journal of Internet Services and Information Security, 2012, Vol. 3, Number 4, pages 134-147.

[9] Hwang T, Shin Y, Son K, Park H. "Design of a Hypervisor-based Rootkit Detection Method for Virtualized Systems in Cloud Computing Environments", AASRI Winter International Conference on Engineering and Technology (AASRI-WIET), 2013, pages 27-32.

[10] Agobot, http://www.f-secure.com/v-descs/agobot.shtml, 2012.

[11] Xiongwei Xie, Weichao Wang. "Rootkit Detection on Virtual Machines through Deep Information Extraction at Hypervisor-level", In Proceedings of IEEE Conference on Communications and Network Security (CNS), 2013, pages. 498–503, National Harbor.

[12] M. Schmidt, L. Baumgartner, P. Graubner, D. Bock, B. Freisleben, "Malware Detection and Kernel Rootkit Prevention in Cloud Computing Cnvironments", Procedings of Int. Euromicro Conf. on Parallel, Distributed and Network-Based Processing, PDP'11, IEEE Computer Society, Washington, DC, USA (2011), PP. 603–610

[13] Musavi, S.A. , Kharrazi, M., "Back to Static Analysis for Kernel-Level Rootkit Detection", Published in: Information Forensics and Security, IEEE Transactions on (Volume:9 , Issue: 9), pages 1465–1476, Sept. 2014.

[14] Kruegel, C., and Toth, T.,"Using Decision Trees to Improve Signature-based Intrusion detection". In Proceedngs of Int'l Symp. Recent Advances in Intrusion Detection, 2003.

[15] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009), "Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges", Computers & Security, 28(1–2), PP. 18–28.

[16] http://www.ossec.net/

[17] X. Wang and R. Karri, "Numchecker: Detecting Kernel Control-flow Modifying Rootkits by using Hardware Performance Counters", Proceedings of DAC, 2013, pages 1-7.

[18] C. Kruegel, W. Robertson and G. Vigna, "Detecting kernel-level rootkits through binary analysis", Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC'04), Tucson, AZ, Dec. 2004.

[19] A. Baliga,V. Ganapathy and L. Iftode, "Detecting Kernel-level Rootkits using Data Structure Invariants", IEEE Transaction on Dependable Secure Computing, Vol. 8, No. 5, pages 670-684, 2011.

[20] T. Garfinkel and M. Rosenblum., "A Virtual Machine Introspection Based Architecture for Intrusion Detection", Proceedings of the Symposium on Network and Distributed Systems Security (SNDSS), pages191-206, February 2003.

[21] https://www.virtualbox.org/

[22] http://packetstormsecurity.com/

[23] https://samsclass.info /123 /123_S11.shtml

[24] Shui Yu ,Yonghong Tian ,Song Guo , Wu, D.O.,"Can We Beat DDoS Attacks in Clouds?", IEEE Transactions on Parallel and Distributed Systems, pages 2245–2254, Sept. 2014

[25] http://sourceforge.net/projects/loic

978-1-4799-7315-6/15/$31.00 ©2015 IEEE