Framework for Secure Virtualized Environment

Mrs. G.Madhu Priya[1], R.Monesha[2]

Department of Computer science and Engineering
Thiagarajar College of Engineering, Madurai, India
Email id - madhupriyag@tce.edu[1], moneshabharath@gmail.com[2]

Abstract- The adoption and diffusion of the cloud are threatened by unresolved security issues. Our proposed work is to protect the integrity of the guest virtual machines and infrastructure components while remaining fully transparent to virtual machines and to cloud users. Various attacks like Denial of services(DOS), Honeypots, password hacking by the hackers affects the availability, integrity and confidentiality of the cloud. The proposed work is to provide a security framework for the cloud. The effectiveness of the system against the attack is tested.

Index Terms- Attacks, Cloud Computing,Security, Virtualization

I. INTRODUCTION

Cloud computing is a model for enabling convenient, on demand network access to share a pool of configurable computing resources, that can be rapidly provisioned and released with minimum management. In cloud there is a significant work load consolidation. Instead of installing all software for each computer, the users have to load one application. That application would allow users to log into a web based service, which hosts all the programs the user would need. Cloud model promotes availability and is composed of essential characteristics.

Cloud computing promotes three delivery models. They are SaaS, PaaS, IaaS. If users don't want any configuration, just want to upload files and leave website to run, then software as a Service is used. Its advantages are easy backups, flexible pricing and portability. A preconfigured operating system but a reasonable amount of control over the rest of the configuration, PaaS is used. Its advantages are minimizing the developers maintenance time while still providing a considering amount of customization and configuration. Virtualization applications can be moved onto different hardware quickly in response increased demand. If users want an operational server with total control over the operating system and configuration, Infrastructure as a Service is used. Its advantage is that the user can effectively have clean installed of the required environment available at all times.

Virtualization, a technique of dividing resources of a computer into multiple execution environments, by applying one or more technologies such as software practioning, time sharing, partial or complete machine simulation, emulation and so on.

The benefits of virtualization are increased server utilization, simplify legacy software migration, host mixed operating systems per physical platform, stream line test and development, isolate software faults, relocate existing partitions, create dedicated or as needed partitions can apply different settings to each partition.

The contribution of our work is to provide a secure virtualized environment for the cloud. The attackers access the configuration files and make the services unavailable, try to illegally enter into the system by hacking the users passwords by recognizing the keystrokes and do active attack by modification of messages. Hence thereby the integrity, confidentiality and availability of the system got affected. The proposed system identifies the users who do malicious actions, and block the user preventing access of the system.

II. RELATED WORK

As virtualization has become more popular, concern over the technology's security has grown [4]. Traditional security techniques often don't work well with virtualization so vendors are trying different approaches. This paper focuses the dynamic property of virtual machines. The primary is since the VM's are created and deleted according to the usage, new VM's are installed automatically so the protection has to be provided systematically. Machines may go for the migration so protection has to be provided in the migrated place. Malware is probably the most significant computer threat to enterprises and businesses [1]. To address the exposure gap left by antivirus products an emerging category desktop security products that use the application sandboxing attempts to address malware threats by containing the malicious behavior. This paper describes how virtualization environment can itself be a trusted sandboxing environment.It describes about a systematic review on the security effects of virtualization[3]. The virtualization technology has a clear positive effect on availability,but the effect on confidentiality and integrity is less positive.Cloud computing adoption and diffusion are threatened by unresolved security issues that affect both the cloud provider and the cloud user[2]. This paper shows how virtualization can increase the security of the cloud computing by protecting the integrity of the guest VM's and the cloud infrastructure components. In particular this paper proposed a novel architecture called Advanced Cloud Protection System(ACPS) and aimed at guaranteeing increased security to cloud resources.Cloud computing is a break through technology that will continue to unleash new innovations and bring new efficiencies and advantage to the business.[6] Enterprises face limitations in using the cloud for high performance and mission critical applications such as ERP. This paper seeks to clear up some misperceptions and help people to make better choices.Analysis technique both use virtual machine and emulation environment provides system restoring capabilities to automate the analysis of malware sample.[5] VM aware malware hides its nature in the system and shows its malicious effects and attacks the system. The work in this paper is the analysis, and evaluation of malware analysis framework for

bare metal systems in a fast and rebootless system.Virtualization is essential to cloud computing yet its security vulnerabilities in the cloud environment have not been sufficiently studied.[9] The analysis of cloud security focuses on how the attacks in virtualization affect different cloud service models. This paper deals with analyzing the nature of three major attacks Spoofing, VM Hopping, VM Mobility.Security is a great issue for many organization to move into the cloud.[10] The main work in this paper is to design new security technologies to cope up with the security challenges of cloud regarding enterprise. The cloud should be provided with security not only at the infrastructure level instead the security should be provided in such a way that it should be beyond infrastructure and into infostructure and metastructure. In this paper they have discussed four technology patterns.Organization fully focuses on securing system from malware and intrusion. Many virtualized systems run on same machine so we need different security level.[7] It distributes the host machine resources to each program running on virtualized OS. Virtualized machines communicate with the help of network connections. This paper discusses about the configuration management problem and virtualization security tools.Virtual machine are software entities emulate machine's functionality. Hypervisor control these machines.[8] hypervisor are categorized into,Directly on top of hardware: Non Hosted Integrated with the host OS: Hosted. The hypervisor over a VM provides a trusted computing base provides intrusion detection, integrity protection and malware analysis. This paper focuses on guarding the hypervisor layer. Malware attackers take a snapshot of a virtual machine and use the copy and pretend like a virtual machine[12]. The work focused here is the detection and mitigation techniques of attacks.They have mentioned an application called VM Safe, it is a compliant security application acts as a virtual machine and provides security to host machine. Protecting users from various attacks. Offers virtual environment the capability to monitor, inspect and filter packets in the hypervisor. It managed servers and virtual network the same level of visibility.

III. IMPLEMENTATION

A. HTACCESS File:

A **.htaccess** (hypertext *access*) file is a directory-level configuration file supported by several web servers. The original purpose of .htaccess reflected in its name was to allow per-directory access control , for example requiring a password to access the content. Nowadays however, the .htaccess files can override many other configuration settings including content type. A .htaccess file is often used to specify security restrictions for a directory, hence the filename "access". The .htaccess file is often accompanied by a .htpasswd file which stores valid usernames and their passwords. Users allow or deny to block other users by IP address or domain. Also, use to block bad bots, rippers and referrers.

B. Hash Values:

A hash function is any well defined procedure or mathematical function that converts a large, possibly variable sized variable sized amount of data into a small datum, usually a single integer that may serve as an index to an array. The values returned by the hash function are called as hash values, hash codes, hash sums, checksums or simply hashes.

Pseudocode: To Compute Hash Value

```
var int[64] r, k
r[ 0..15] := {7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22}
r[16..31] := {5,  9, 14, 20, 5,  9, 14, 20, 5,  9, 14, 20, 5,  9, 14, 20}
r[32..47] := {4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23}
r[48..63] := {6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21}

//Use binary integer part of the sines of integers (Radians) as constants:
for i from 0 to 63
    k[i] := floor(abs(sin(i + 1)) × (2 pow 32))
end for
//(Or just use the following table):
k[ 0.. 3] := { 0xd76aa478, 0xe8c7b756, 0x242070db, 0xc1bdceee }
k[ 4.. 7] := { 0xf57c0faf, 0x4787c62a, 0xa8304613, 0xfd469501 }
k[ 8..11] := { 0x698098d8, 0x8b44f7af, 0xffff5bb1, 0x895cd7be }
k[12..15] := {0x6b901122,0xfd987193,0xa679438e, 0x49b40821 }
k[16..19] := { 0xf61e2562,0xc040b340,0x265e5a51, 0xe9b6c7aa }
k[20..23] := { 0xd62f105d,0x02441453,0xd8a1e681, 0xe7d3fbc8 }
k[24..27] := { 0x21e1cde6,0xc33707d6,0xf4d50d87, 0x455a14ed }
k[28..31] := { 0xa9e3e905, 0xfcefa3f8, 0x676f02d9, 0x8d2a4c8a }
k[32..35] := { 0xfffa3942, 0x8771f681, 0x6d9d6122, 0xfde5380c }
k[36..39] := {0xa4beea44, 0x4bdecfa9, 0xf6bb4b60, 0xbebfbc70 }
k[40..43] := { 0x289b7ec6,0xeaa127fa, 0xd4ef3085, 0x04881d05 }
k[44..47] := { 0xd9d4d039,0xe6db99e5, 0x1fa27cf8, 0xc4ac5665 }
k[48..51] := {0xf4292244, 0x432aff97,0xab9423a7, 0xfc93a039 }
k[52..55] := {0x655b59c3, 0x8f0ccc92, 0xffeff47d, 0x85845dd1 }
k[56..59] := {0x6fa87e4f, 0xfe2ce6e0, 0xa3014314, 0x4e0811a1 }
k[60..63] := {0xf7537e82,0xbd3af235, 0x2ad7d2bb, 0xeb86d391 }

//Initialize variables:
var int h0 := 0x67452301   //A
var int h1 := 0xefcdab89   //B
var int h2 := 0x98badcfe   //C
var int h3 := 0x10325476   //D

//Pre-processing: adding a single 1 bit
append "1" bit to message
/* Notice: the input bytes are considered as bits strings,
```

where the first bit is the most significant bit of the byte.[37]

```
//Pre-processing: padding with zeros
append "0" bit until message length in bit ≡ 448 (mod 512)
append length mod (2 pow 64) to message

//Process the message in successive 512-bit chunks:
for each 512-bit chunk of message
   break chunk into sixteen 32-bit words w[j], 0 ≤ j ≤ 15
//Initialize hash value for this chunk:
   var int a := h0
   var int b := h1
   var int c := h2
   var int d := h3
//Main loop:
   for i from 0 to 63
      if 0 ≤ i ≤ 15 then
         f := (b and c) or ((not b) and d)
         g := i
      else if 16 ≤ i ≤ 31
         f := (d and b) or ((not d) and c)
         g := (5×i + 1) mod 16
      else if 32 ≤ i ≤ 47
         f := b xor c xor d
         g := (3×i + 5) mod 16
      else if 48 ≤ i ≤ 63
         f := c xor (b or (not d))
         g := (7×i) mod 16
      temp := d
      d := c
      c := b
      b := b + leftrotate((a + f + k[i] + w[g]) , r[i])
      a := temp
   end for
//Add this chunk's hash to result so far:
   h0 := h0 + a
   h1 := h1 + b
   h2 := h2 + c
   h3 := h3 + d
end for

var char digest[16] := h0 append h1 append h2 append h3
//(Output is in little-endian)

//leftrotate function definition
leftrotate (x, c)
   return (x << c) binary or (x >> (32-c));
```

C. *Cryptographic Hashing:*

Used for data verification, user verification and authentication. A strong cryptographic hash function has the property of being very difficult to reverse the result of the hash and hence reproduce the original piece of data. Cryptographic hash function are used to hash users passwords and have the hash of passwords stored on the system rather than having the password it stored. Cryptographic hash values are also seen as irreversible compression functions being able to represent large quantities of data with a single ID in which they are useful in seeing whether or not the data has been tampered with any other illegal perspectives.

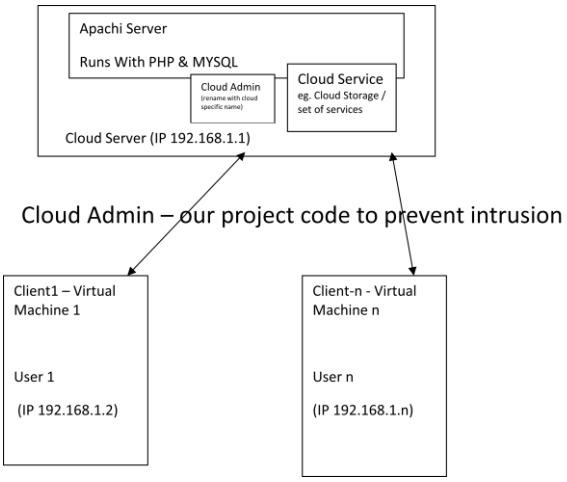

Fig1 : Working model of the system

IV. DETECTION OF ATTACK

A. *Providing Integrity by Detecting The Data Modification Attack:*

If the attacker tries to access the .htaccess file which is the configuration file for the Apache server and if the attacker modifies those configuration files for abnormal memory usage and connection attempts, the services in which the server provides will be unavailable to the legitimate users. The proposed system calculates the hash values for all the configuration files and stores the hash values in the hash DB. With the help of hash values we can find the integrity of those files has been affected or not. Every time the system compares the hash values and if the already computed hash values and the current hash values are different the integrity is affected. If the hash values are not different then the integrity has not been affected. Once the system finds that the integrity has been affected the system detects which user modifies the files and once he logs on the alert will be sent and he is stopped to use the services. The system has a Warning detector reports the status of the data and stores the status in the warning pool. The components that are monitored are library files and configuration files.

B. *Providing Confidentiality Through Keystroke Timing*

Keystroke dynamics is part of larger class of biometrics known as behavioral biometrics. Their patterns are statistical in nature. The reality here is that behavioral biometrics use a confidence measurement instead of traditional pass or fail measurements. The system calculates the keystroke timing for the users passwords. Once the user log again it verifies the keystroke of the corresponding password. If both the timing matches the user is allowed to access the virtual machine services.

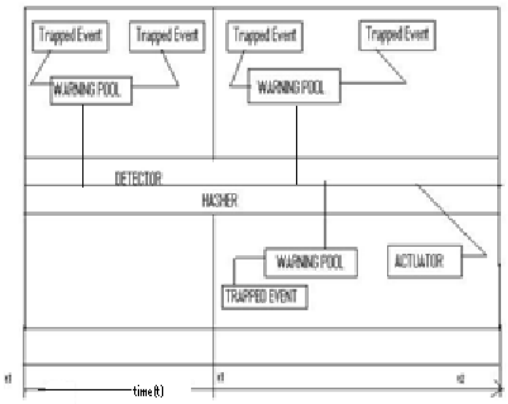

Fig 2: Timing Analysis On The Attack Detection

C. *Detection of Attacker:*

If the user of the virtual machine tries to access or modify the unauthorized files, the alerts will be sent to the administrator by the system, the IP address from which the attacker attacks the files will be stored in the warning detector. The system blocks the user and alerts him once the user logs on into the system. The system replaces the affected files.

V. EFFECTIVENESS

The detection capabilities of our system are assessed against known attack techniques. However, since source code for many attacks is not publicly available, we performed our test by simulating attack steps. The system is effective to maintain confidentiality of the individual users files through the keystroke timing, and detects the modification of data thereby the integrity is preserved.

VI. CONCLUSION

The system prevents the attacker to stop the server and it detects the attacker who modifies the configuration files and makes the services unavailable and it also provides confidentiality by preventing him from illegal connection attempts to the services. Hence thereby the integrity, confidentiality and availability of the services is preserved.

REFERENCES

[1] Chris Greamo and Anup Invincea ," Sandboxing and Virtualization Modern Tools For Combating Malware" , IEEE Computer and Reliability Societies,Vol.19, No.79 to 83,2011.

[2] Flavio Lombardi and Roberto Di Pietro," Secure virtualization for Cloud Computing", Journal of Network and computer Applications,Elsevier,vol.17,no.5,pp.403-418, 2010 .

[3] Andre Van Clee,Wolter Pieters and Roel Wieringa, " Security Implications of Virtualization: A Literature Study",International Conference on Computational Science and Engineering,IEEE, vol3, no.353to358,2009 pp.15- 24,2008.

[4] Le Garber "The Challenges of Securing the Virtualized Environment" IEEE Transactions on parallel and Distributed Systems,vol.16, no.9, pp.799-813,september,2012.

[5] Dilung Kirat,Giovanni Vigna,Christopher Krueigei, "Barebox:E_cient Malware Analysis On Baremetal",IEEE/ACM Transactions on Networking,vol.9, no.3, pp.226-237, June,2011.

[6] Paul Homann,Daan Woods,"Cloud Computing:The Limits of Public Cloud For Business Application",Published by IEEE Internet Computing,vol.7, no.4, pp.162-164,April 2010.

[7] Steven J.Vaughan Nichols,"Virtualization Sparks Security concerns",Published by IEEE security and privacy,vol.19,no.4, April 2008.

[8] O.Sami Saydjari , "Taming Virtualization",Published by IEEE Security and Privacy ,2008.

[9] Hsinyi Tsai,Melanie siebenhaar anan Huang,Ralf steinmetz,"Threat as a service? Virtualization impact on cloud security", published by IEEE Security and Privacy , vol.16, no.6, December 2010.

[10] John Peter,Gunar Peterson,"Don't trust and verify A Security Stack Architecture For The Cloud", Published by IEEE Security and Privacy,pp.1223- 1228, 2010.

[11] Lori F.Kaufman,"Can A Trusted Environment Provide Security",IEEEcomputer society,pp.V2-67 to V2-71,2010.

[12] Matthew carpenter,Tom Liston,ED Skoudis,"Hiding virtualization from Attackers and Malware",IEEE Security and Privacy ,pp.804-809,2007.