International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 13 (2017) pp. 4035-4043

Building Secretkey Based Virtual Machines in Cloud Computing to avoid Vulnarabilities in Hypervisors

A. Praveen[1], Dr. M. Nagalakshmi[2], G.Sreenivasa Yadav[3] and M.Thukaram Reddy[4]

[1]*Associate Professor, Department of IT, Institute of Aeronautical Engineering, Hyderabad, India.*

[2]*Associate Professor, Department of CSE, MLR Institute of Technology, Hyderabad, India.*

[3]*Assistant Professor, Department of CSE, SNIST Hyderabad-501301, India.*

[4]*Assistant Professor, Department of IT. BVRIT, Narasapur, Hyderabd, Telangana, India.*

[1]*ORCID: 0000-0002-8807-0964,* [3]*ORCID: 0000-0002-4797-5636*

Abstract

Cloud computing becomes more and more adaptive technology for the current computing world. In the cloud environment to provide infrastructure resources to the end customer they are using virtualization technology. Virtualization is built on a software module called hypervisors. By implementing virtualization in the cloud the same resources are shared among different end customers. This may lead to accessing of one end user account to another end user account. This is because of security problems in the hypervisors. In this paper we are proposed a secret key based virtual machine by generating a secret key using symmetric key generation algorithms to access the virtual machine to avoid vulnerabilities in the Hypervisors. Here we build a system to generate a unique secret key for each user in the virtual machine when the user uses the cloud resources through virtual machine and the virtual machine allow the end user to access the resources by checking the secret key.

Keywords: Cloud Computing, Virtualization, Hypervisors, secret key, vulnerabilities.

INTRODUCTION

Now a days cloud computing becomes an emerging technology for all the areas in the computing world. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models [1].Nodes that are implemented with Cloud and virtualization are more vulnerable to cyber attacks when compared to normal nodes because of their size and complexity. Cloud provides three basic service models like Infrastructure as a Service to provide Hardware Infrastructure to the end user, Platform as a Service to provide platform to different applications to run on the cloud and Software as a Service to provide direct applications to the end users.

To provide the resources efficiently to the end customers cloud computing is implemented on the virtualization platform. Virtualization is a model that refers the partitioning of physical resources into virtual parts. Virtualization provides cost effective utilization of resources. Virtualization is mainly implemented in different areas like Networking, Storage, Server, Data, Desktop and Applications are virtualized. There are different approaches to implement virtualization technology like Hosted Operating system and Native Hypervisor or Bare Metal virtualization.

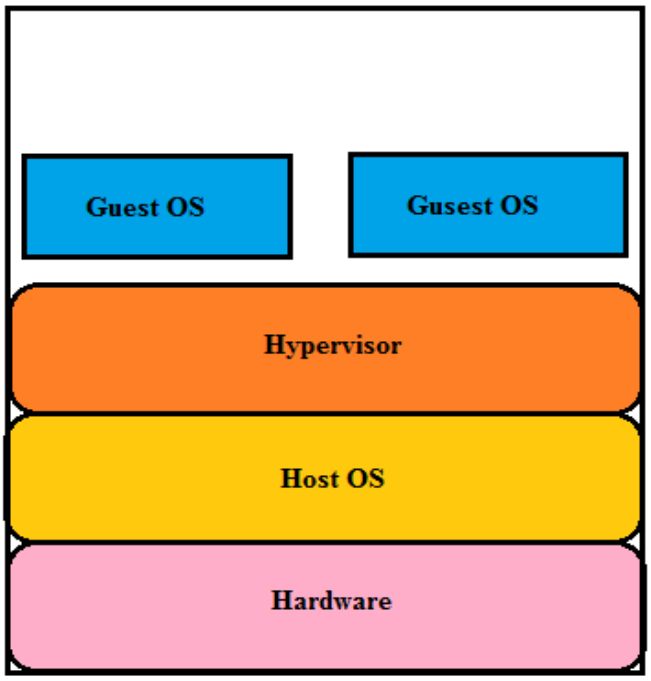

Figure 1: Hosted Operating System Virtualization

Fig 1 describes the Hosted Operating System Virtualization. Here the Host operating system directly sits on the Hardware and the Hypervisor is installed on the Host operating system and on the hypervisor the guest operating systems are installed. Fig 2 shows the implementation of Bare Metal virtualization. Here hypervisor is directly installed on the Hardware and the guest operating systems are placed on the Hypervisor. Bare Metal Hypervisor has direct access to the hardware resources; it has greater scalability, good performance and robustness when compared to Hosted Operating system virtualization.

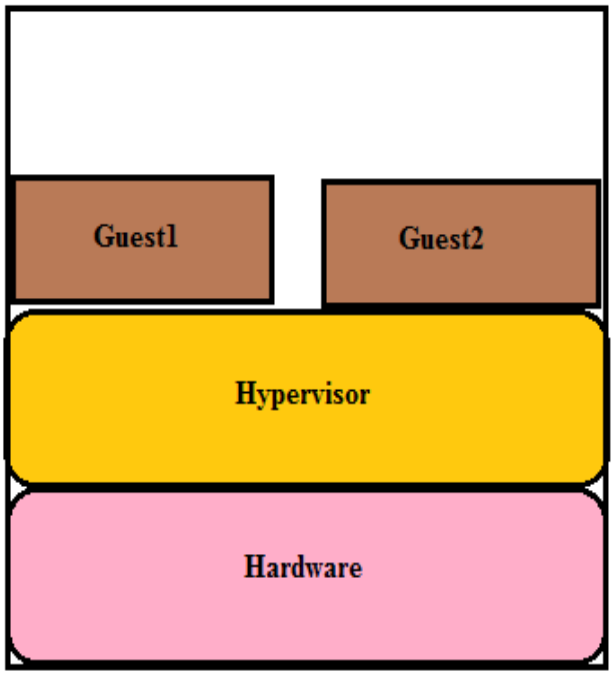

Figure 2: Bare Metal Virtualization

Virtual Machines on the Internet are bare different kinds of communications that virtualization technology can help filtering while assuring a higher degree of security. In particular, virtualization can also be used as a security component; for instance, to provide monitoring of VMs, allowing easier management of the security of complex cluster, server farms, and cloud computing infrastructures to cite a few. Here Virtualization allows providing more security in the cloud.

The main aim of this paper is to analyze cloud security issues and model, threats and proposing a new protection system for the cloud in virtualization. The remainder of this document is organized as follows: next section surveys related work. Section 3 provides cloud security issues while Section 4 describes Protection System implementation finally Section 5 describes some conclusions.

LITERATURE SURVEY

Pearson (2009) [2] is talking in depth about privacy issues in cloud computing, nominal cloud security issues are discussed in the literature review of Gu and Cheung (2009)[3]. Siebenlist (2009)[4] has discussed some interesting features in cloud security issues, Cachin at al(2009)[5] made a complete survey on security in the context of storage services in the cloud . An exhaustive assessment has been made by Enisa(2009) [6] in the area of cloud security assessment. Armbrust et al. (2009) [7] made a worthy survey on cloud computing, Mukundha[2016][8] has discussed some of the security problem in the cloud security. All the above papers have been the starting points of our work and we refer to them in terms of problems and terms definition. A fundamental reference for our research is the work on co-location (Ristenpart, 2009)[9] by Ristenpart. It describes that is it a possible way to instantiate an increasing number of guest Virtual Machines until one is placed co-resident with the target Virtual Machine. Once co residency is successfully achived, then attacks can extract information from a target Virtual Machine on the same machine. An attacker might also actively trigger new victim instances exploiting cloud auto-scaling systems. Ristenpart shows that it practical to hire additional VMs whose launch can produce a high chance of co-residence with the target Virtual Machine. He also shows that determining co-residence is quite simple.

Majority of the current existing intrusion detection and integrity monitoring solution can be applied successfully on the cloud computing. File system Integrity Tools and Intrusion Detection Systems such as Tripwire (Kim and Spafford,1994)[10] and(AIDE) (AIDE team, 2005)[11] can also be deployed in virtual machines, but are exposed to attacks possibly coming from a malicious guest machine user. Furthermore, when an attacker identifies that the targeted system is existed in a virtual environment ,it may attempt to breakout of the virtual environment through the vulnerabilities that are existing in the virtualization(very rare at the time of writing Secunia, 2009)[12] in the Virtual Machine Monitor(VMM). Most present approaches leverage VMM isolation properties to secure VMs by leveraging various levels of virtual introspection. Virtual introspection (Jiang etal.,2007)[13] is a process that allows to observe the state of aVM from the VMM. SecVisor (Seshadri etal., 2007)[14] Lares (Payne etal.,2008)[15] and KVM-L4 (Peter etal.,2009)[16], to name a few, leverage virtualization to observe and monitor guest kernel code integrity from a privileged VM or from theVMM. Nickle (Riley etal. 2008)[17] aims at detecting kernel root kits by monitoring the integrity of kernel code. However, Nickle does not protect against kernel data attacks (Rhee etal.,2009)[18], where asour solution does.Most proposals have limitations that prevent them from being used in distributed computing scenarios(e.g.. SecVisor only supports one guest per each host)or just do not consider the special requirements or peculiarities of

distributed systems; for instance, KVM-L4 shares the same underlying technology as Lombardi and DiPietro(2009)[19] but the additional context switching overhead in the64-bit scenario, representing the vast majority of cloud hosts, remains to be verified. Also worth citingare IBM on (Ranadive etal., 2009)[20], a monitoring utility using introspection for asynchronous monitoring of virtualized network devices, and Low Grid (Salza etal.,2006)[21], an example of autonomic reaction system.

In an effort to make nodes resilient against long-lasting attacks, Self-Cleansing Intrusion Tolerance (SCIT) (Huang etal., 2006)[22] treats all servers as potentially compromised(since undetected attacks are extremely dangerous overtime). SCIT restores servers from secure images on a regular basis. The drawback of such a system is that it does not support long-lasting sessions required by most cloud applications .Similarly, VM-FIT (Distler etal.,2008)[23] creates redundant server copies which can periodically be refreshed to increase the resilience of the server. Finally, Sousa etal.(2007)[24] approach combines proactive recovery with services that allow correct replica store act and be recovered when there is a sufficient probability that they have been compromised. Along with the many advantages brought by virtualization, there are additional technological challenges that virtualization presents, which include an increase in the complexity of digital forensics (Pollitt etal., 2008)[25] investigations as well as questions regarding the forensics boundaries of a system.

CLOUD COMPUTING SECURITY ISSUES

Main drawback of the cloud computing is the loss of control on the data that is the end user does not know where exactly the data is stored and where the data is processed in the cloud. Data and computation are mobile and can be shifted to the systems the end user cannot control directly. Cloud computing uses the internet, data in the internet can easily cross the international borders it leads to another security threat. The dark side of Infrastructure as a service is that the cloud provider is gets paid for running a service he does not now the details of it. To update misuse of data problems tend to be regulated by a service agreement, where such type of contract should be enforced and controlled by monitoring tools. (Haeberlen, 2009)[26].Accessing of sensitive information has to limited to a subset of privileged clients. In the cloud same type of instance is used by more number of customers so it is very important and very difficult to separate each customer data. It is possible an attacker attack the data and spoil sensitive data stored on the cloud. Recovering the loss data is also another big challenge to the cloud providers when the disaster occurs. It is very difficult to trace cloud service accountability, even though it is necessary for some times.

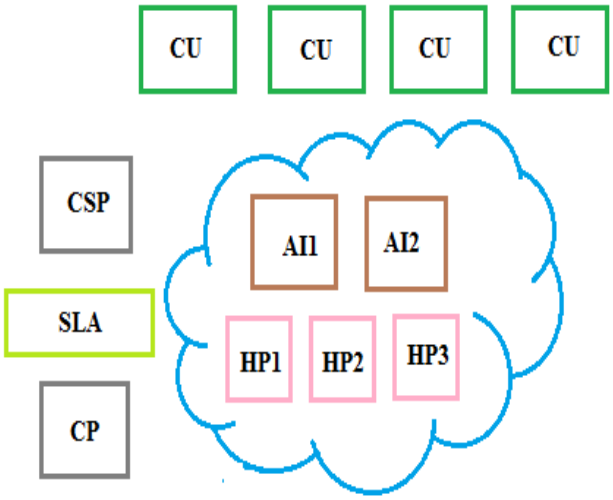

Figure 3: Cloud Service Model

The about Fig3 gives the working of cloud services. Cloud Provider (CP) provides the Cloud infrastructure to the Cloud Service Provider (CSP). In between CP and CSP there is some Service Level Agreements (SLA) to provide quality services. Host Platform (HP) is a machine to deploy applications. Application instance (AI) is applications that are running on the Host machines to service end customers and the lastly the Cloud users (CU) access the resources that are existing in the cloud in the form of Application Instances.

SYMMETRIC KEY GENERATION ALGORITHMS

Symmetric key generation algorithm is a cryptographic key generation algorithm to generate a single key to encrypt the data in the transmission. By using the cryptographic symmetric key techniques in the virtualization layer we provide more security in the cloud virtualization. Symmetric key is generated by using many symmetric key generation algorithms like AES, 3DES, Two Fish, Serpent, Blowfish, CAST5, RC4, Skipjack, IDEA etc. The keys that are generated by the symmetric key algorithm must be known by the end user on the virtual machine and the same key is stored in the User key management system for checking the validity of the end user to access the resources in the cloud through the cloud and these are commonly known as secret keys. Here a secret key is generated by one or more entities of the key sharing resources or from a trusted third party that is intended to share entities in a secure manner. These Symmetric keys are used to generate message authentication codes like passwords, encrypt or decrypt data in an appropriate mode or to derive the additional keys.

The Symmetric keys are generated by using different ways in cryptography. They are:

- Direct Generation of Symmetric keys

- Key Agreement schemes
- Pre-Shared Keys
- From Passwords
- By Combining Multiple keys and other data

Direct Generation of symmetric key is generated by the output of an RGB it is specified in (Sp 800-57-1)[27].key generating agreement is available with the key generating entities and a symmetric key is established with another entity that has the same capabilities; this process avoids the sharing of symmetric in between the two entities.

Fig 4 describes the typical key agreement process. A shared secret key is given to a key generation method to generate keying material. Based on DLC approved key management methods are specified (SP 800-56A)[28] and in (SP 800-56B)[29] specified approved key management methods based on IFC. In (SP 800-56C)[30] Approved expansion and Extraction procedures are specified. The above all schemes are used in key-agreement schemes. The maximum security with this process is depending on 1) The strength of asymmetric keys 2) Method of key derivation 3) derived key length and 4) the algorithm that is used with the derived keys.

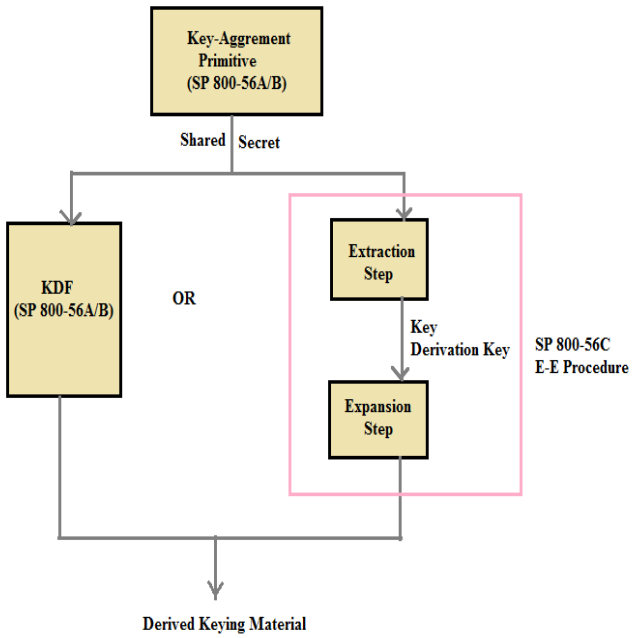

Figure 4: Key Agreement Process

In Pre-Shared key process the symmetric keys are derived by using a key derivation function. Pre shared key may be generated by using a approved RBG and distributed, agreed upon using key agreement scheme or with the key derivation function. To generate a symmetric keys by using passwords are provided a sort of approved methods for storage applications (SP 800-132)[31] for these type f applications it is strongly recommended that the select passwords should have very large amount of entropy. By combining the symmetric keys with the key generating module and the other data items that are independent of keys is possible to generate symmetric keys. In this process we are using a well know symmetric key generation algorithm called Advanced Encryption Standards (AES) algorithm.

Algorithm for Symmetric Key Generation:

Given

- N, the required number of pseudorandom bytes
- U, an optional user-supplied seed consisting of an arbitrary number of bytes
- L, P, two 64-bit continuous check values stored in thread-safe memory

Step 1.

Generate 32 pseudo-random bytes with the seed key generator, adding the user-supplied seed, U, if any.

Step 2.

Set the 192-bit Triple DES key, K, as the first 24 bytes generated in step 1, and set the seed, S, as the last 8 bytes.

Step 3.

Set D as a 64-bit representation of the current date and time.

Step 4.

Generate the 64-bit block X0 = G(S, K, D) where G is the 64-bit block generator and S is updated as per algorithm.

Step 5.

Set up to carry out continuous random number generator tests:

1. If X0 equals L or X0 equals P, stop and notify a failure of the continuous random number generator test.
2. Set L = X0 and store in thread-safe memory for the next call.
3. Set P = X0.

Step 6.

For R = N until R is equal to zero, do:

1. Generate a 64-bit block X = G(S, K, D), updating S in the process.
2. If X equals the previously-generated block, P, then stop and notify a failure of the continuous random number generator test.
3. Set B = the lesser of R and 8.

4. Output B bytes from X.
5. Set R = R - B.
6. Set P = X.

Step 7.

Generate a final block Xf = G(S, K, D) and set P = Xf.

Step 8.

Zeroise K, S, D, X and any other internal buffers used. Retain L and P for subsequent use.

64-bit Generator(G)

Given

- D, a 64-bit representation of the current date-time
- S, a secret 64-bit seed that will be updated by this process
- K, a secret Triple DES key

Step 1. Compute the 64-bit block X = G(S, K, D) as follows:

1. I = E(D, K)
2. X = E(I XOR S, K)
3. S' = E(X XOR I, K)

Where E(p, K) is the Triple DES encryption of the 64-bit block p using key K.

Step 2. Return X and set S = S' for the next cycle.

SYSTEM MODEL

In the proposed system we build a secret key management system for each virtual machine created in the cloud environment.

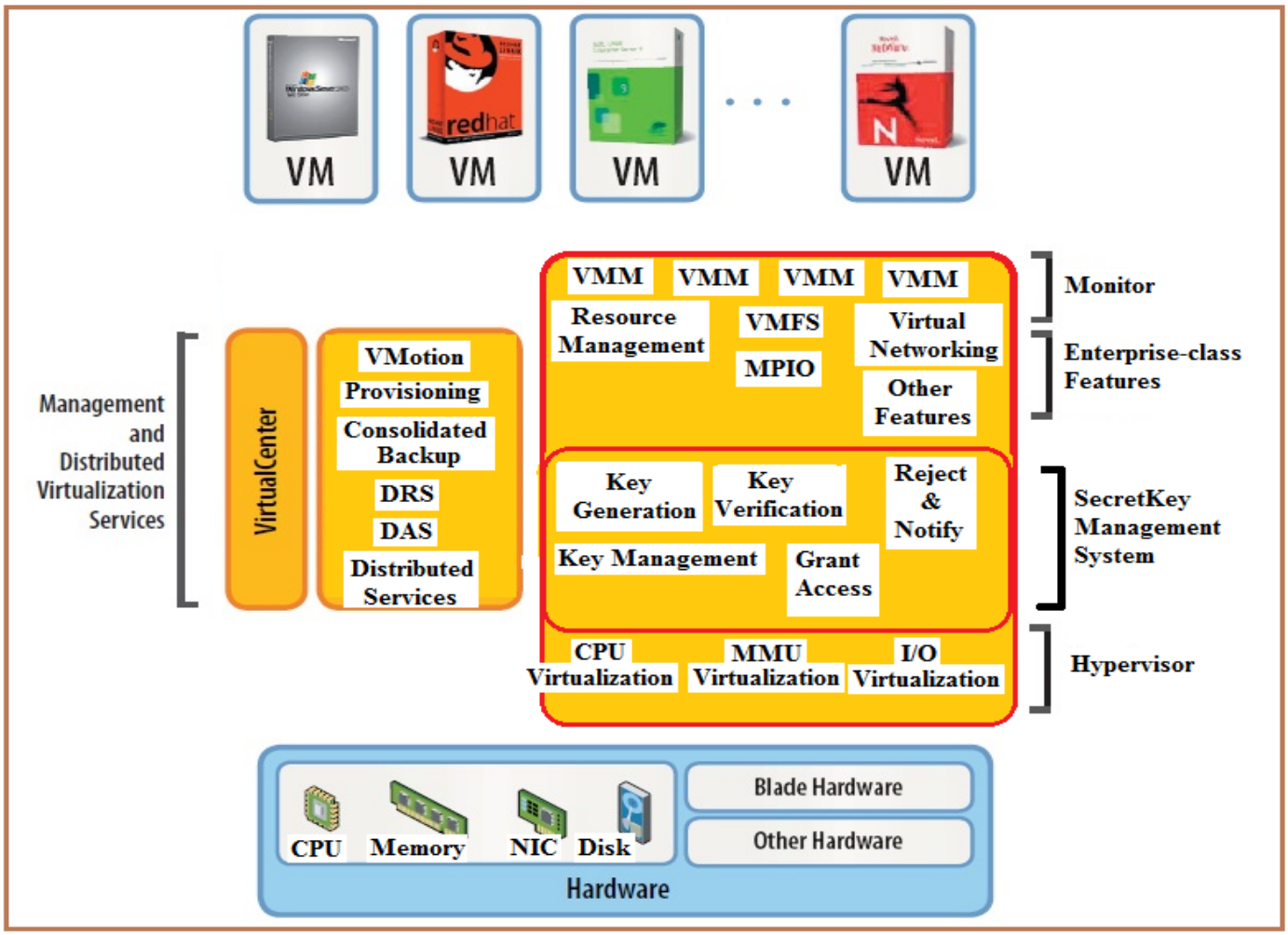

Figure 5: Proposed Secret Key Management System

Fig 5 represents the proposed Secret key Management system in the Hypervisors used in virtualization. Here top level contains the virtual machines created with different instances middle layer contains the virtualization environment and it contains different services like Monitor, Resource Management, Secret Key Management, Hypervisors and Distributed Services. Virtual Machine Monitoring (VMM) monitors the usage of resources like automatic resource provisioning, High availability and workload migration support.

Resource Management is another service to properly organize the service of CPU, Memory, Disks and Networking components. Multipath Input and output (MPIO) is a Microsoft framework to provide failovers data transfers between storage devices and Microsoft systems. Virtual Machine File System (VMFS) is a cluster file system to facilitate virtualization for storing different virtual machines in the virtual environment. Virtual Networking provides the networking environment for the virtual machines created in the cloud environment.

Secret Key Management system is a model to avoid vulnerabilities in virtual machines in cloud computing. Here Key Generator system generates a unique symmetric random key by using AES symmetric key algorithm for each virtual machine that is created in the cloud computing for each user. The key verification system checks the key of a virtual machine and the user using the virtual machine in the same session. If key is matched with the existing key in the virtual key management system and the key with the virtual machine then it provides the access of virtual machine otherwise it rejects the access of virtual machine and report to the Resource manager to block the resource allocation. In the last layer of hypervisor it implements the virtualization of core modules like CPU,I/O and Memory modules.

ATTACK MODEL

The Attack model we use in the discussion is based on one of the scenarios utilized by Prasad saripalli [32] to expose data leakage in Amazon EC2 public Cloud. The behavior of data Security for a defined vulnerability is that there could be a more number of attacks to exploit it [32]. On one successful attack against a system will analyze most of the possible vulnerabilities that can be utilized. These attacks vary in the sense of their behavior; for example, it is easy to identify any distributed denial of service (DDoS) attack and any attack consisting of port scanning due to the sudden increase in traffic of the cloud. Also, it is easy to identify viruses due to their unique signatures; whereas it is very complex thing to detect iFrame attacks. An iFrame attack is an attack where an HTML code is inserted inside another HTML document as a frame in order to collect credit card information for instance.

The following are the list of attacks that could be used efficiently over the Cloud infrastructure:

- **Side Channel attack:** a side channel attack is any type of attack depending on data gained from the physical implementation of a system. There is different type of side channel attacks known in the field; some of the well known side channel attacks are power consumption attacks, timing attacks, and differential fault analysis.
- **Brute Forcing:** brute forcing is an attack mechanism applied over any type of attacks. This is one of the easiest methods in order to build an attack but yet it is one of the most common used methods. For example if an attacker wants to find out a password of a system by utilizing a brute force method, the attacker will try every possible combination until the correct password is found. Therefore, brute forcing can be stated as running an attack operation multiple times until a successful breach is achieved. Brute forcing is find as one of the most used top ten attacks by the Data Breach Investigations Report (DBIR) where it forms 22% of data breach attacks [33]
- **Network Probing:** It is a method used to find out the physical topology of a network that consists of servers and IPs connected in the network. Such data can be finding to identify possible targets and to implement an attack for a sub group in the network.

RISK APPROACHES

Virtualization is a risk factor then we need to choose one of the well-known risk management techniques in order to manage and minimize its bump. These strategies are as follows:

1. Eliminating the risk.
2. Mitigating the risk.
3. Transferring the risk.
4. Accepting the risk.

Although avoiding the risk considered as the most efficient methodology, As stated previously most of the benefits are linked with virtualization, so it is not possible to apply it on virtualization.

To avoid risk of virtualization we need to remove implementation of virtualization in the cloud. As defined earlier cloud computing is result of combination of the techniques like virtualization and resource sharing. In order to eliminate the virtualization we have to eliminate resource sharing mechanism in the cloud computing. In either case it is not possible because most of the cloud computing advantages are linked with these techniques.

Virtualization allows the benefit of resource sharing mechanism. This case is valid because of the side channel

attacks defined as taking the benefit of the physical characteristics of the system. By the nature of side channel attacks unlimited in the number and with time it not possible to eliminate all the attacks. However handling the know side channel attacks is very difficult because of their different form of side channel attacks. As discussed by Francisco Rocha et al [34] where they propose a system to deal with shared memory attacks and they reached goal but it is a dependent on hypervisor and it works only on Xen hypervisor.

Another type of attack is timing side channel attack and Peng Li et al [35] proposed a method to eliminate three forms of side channel attacks. If they eliminate all timing side channel attacks but it occupies 2/3 of cloud service provider infrastructure and the solution provided will not avoid all timing side channel attacks. By considering above two examples it is very expensive and hard to eliminate all the side channel attacks. In some cases two types of side channel attacks have relationship where not possible to avoid in the same time that is if the side channel attack A is eliminated then the side channel attack of B is not possible to eliminate. If attack A is avoided means B is Success. So we cannot eliminate side channel attacks by eliminating major features of virtualization and resource sharing. So eliminating the risk method is not possible and it is not applicable when it deals virtualization.

Second method is mitigating the risk here we try to balance between advantage offered by virtualization, security and other factors like performance and cost. Figure 5 describe about without loss of any benefits of virtualization we avoid a risk. Here first it allows the risk by the means of attacker and it block the attacker in the control layer by checking their security key of the virtual machine and the user what they accessed the database. Third method is transferring the risk from one CSP to another CSP having risk handling capabilities. But it is not a fair mechanism to handle the risk. Because no one accepts that the things have a risk. Last method is allowing the risk to provide virtualization for cloud computing applications. In this paper we provide a solution to the virtualization problem by implementing secure key mechanism using Software Defined Network Layers.

CONCLUSION

Virtualization is the asset of the cloud computing providers, even though it comes with security risks. If we concern security is the first concern then eliminate the risk of virtualization. If we avoid this feature in the cloud then the cost of system increase and it will leads to performance degradation with low level of resource utilization. Multitenancy is an opportunity must be implemented without considering any security risks. In these extremes cloud providers provide multitenency to the customer without giving any solutions and at least mitigating the problem to the customers. This type of exposure to the customers leaves from cloud providers.

In this paper we proposed a model for avoiding the virtualization vulnerabilities by using a secure key implementation by symmetric key generation. An approach is discussed to tackle virtualization in both security and resource allocation techniques and it introduces a security as a requirement by implementing resource allocation without affecting performance, cost and power consumption. Finally we are not applying this method in any of cloud models; it is a proposed model for avoiding virtualization issues.

REFERENCES

[1] Computer Security, NIST Cloud Computing definition, special publication 800-145 (September - 2011) https://www.nist.gov/programs-projects/cloud-computing

[2] Pearson S. Taking account of privacy when designing cloud computing services. In CLOUD ’09: Proceedings of the 2009 ICSE workshop on software engineering challenges of cloud computing, IEEE Computer Society, Washington, DC, USA, 2009. pp. 44–52.

[3] Gu L, Cheung S-C. Constructing and testing privacy-aware services in a cloud computing environment: challenges and opportunities. In Internetware ’09: Proceedings of the first Asia-Pacific symposium on internetware. ACM New York, NY, USA, 2009. pp. 1–10.

[4] Siebenlist F. Challenges and opportunities for virtualized security in the clouds. In SACMAT ’09: Proceedings of the 14th ACM symposium on access control models and technologies, ACM, New York, NY, USA, 2009. p. 1–2.

[5] Cachin C, Keidar I, Shraer A. Trusting the cloud. SIGACT News 2009;40(2):81–6.

[6] Enisa. Cloud computing risk assessment. /http://www.enisa.europa.eu/act/rm/ files/deliverablesS, 2009.

[7] Armbrust M, Fox A, Griffith R. Above the clouds: A Berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, February 2009.

[8] Chinthagunta Mukundha, A Survey on Cloud Computing Security Issues and Proposed Associated Solutions, in International Journal of Computer Networks and Security, Recent Science publications, June 2016,pp:1409-1414.

[9] Ristenpart T, Tromert E, Shacham H, Savage S. Hey,

you, get off of my cloud: Exploring information leakage in third-party compute clouds. In CCS '09: Proceedings of the 14th ACM conference on computer and communications security, ACM, New York, NY, USA, 2009. p. 103–15.

[10] Kim GH, Spafford EH.. The design and implementation of tripwire: a file system integrity checker. In CCS '94: Proceedings of the 2nd ACM conference on computer and communications security. ACM, New York, NY, USA, 1994. pp. 18–29.

[11] AIDEteam. Advanced intrusion detection environment/http://sourceforge.net/ projects /aideS, November2005.

[12] Secunia. Secunia advisory. /http:// secunia.com/advisories/36389S, 2009.

[13] Jiang X, Wang X, Xu D. Stealthy malware detection through vmm-based ''out-of- the-box'' semantic view reconstruction. In CCS '07: Proceedings of the 14th ACM conference on computer and communications security. ACM, New York, NY, USA, 2007. pp. 128–38.

[14] Seshadri A, Luk M, Qu N, Perrig A. Secvisor: a tiny hypervisor to provide lifetime kernel code integrity for commodity oses. In SOSP '07: Proceedings of twenty- first ACM SIGOPS symposium on operating systems principles, ACM, New York, NY, USA, 2007. p. 335–50.

[15] Payne BD, Carbone M, Sharif M, Lee W. Lares: An architecture for secure active monitoring using virtualization. In SP '08: Proceedings of the 2008 IEEE symposium on security and privacy (sp 2008), IEEE Computer Society, Washington, DC, USA, 2008. pp. 233–47.

[16] Peter M, Schild H, Lackorzynski A, Warg A. Virtual machines jailed: virtualization in systems with small trusted computing bases. In VDTS '09: Proceedings of the 1st EuroSys Workshop on virtualization technology for dependable systems, ACM, New York, NY, USA, 2009. p. 18–23.

[17] Riley R, Jiang X, Xu D. Guest-transparent prevention of kernel rootkits with vmm- based memory shadowing. In RAID '08: Proceedings of the 11th international symposium on recent advances in intrusion detection, Springer-Verlag, Berlin, Heidelberg, 2008. p. 1–20.

[18] Rhee J, Riley R, Xu D, Jiang X. Defeating dynamic data kernel rootkit attacks via vmm-based guest-transparent monitoring. Availability, Reliability and Secur- ity, 2009. ARES '09. International Conference on, 2009.

[19] Lombardi F, Di Pietro R. A security management architecture for the protection of kernel virtual machines. In TSP '10: Proceedings of the Third IEEE international symposium on trust, security and privacy for emerging applications (to appear), IEEE Computer Society, Washington, DC, USA, 2010.

[20] Ranadive A, Gavrilovska A, Schwan K. Ibmon: monitoring vmm-bypass capable infiniband devices using memory introspection. In HPCVirt '09: Proceedings of the 3rd ACM workshop on system-level virtualization for high performance computing, ACM, New York, NY, USA, 2009. p. 25–32.

[21] Salza S, DiCarlo Y, Lombardi F, Puccinelli R. Leveraging the grid for the autonomic management of complex infrastructures. In: GCA grid computing and applications conference proceedings, 2006. p. 32–7.

[22] Huang Y, Arsenault D, Sood A. Closing cluster attack windows through server redundancy and rotations. In CCGRID, 2006. p. 21.

[23] Distler R, Kapitza R, Reiser HP. Efficient state transfer for hypervisor-based proactive recovery. InWRAITS '08: Proceedings of the 2nd workshop on recent advances on intrusiton-tolerant systems. ACM, New York, NY, USA, 2008. pp. 1–6.

[24] Sousa P, Bessani AN, Correia M, Neves NF, Verissimo P. Resilient intrusion tolerance through proactive and reactive recovery. Pacific Rim International Symposium on Dependable Computing, IEEE 2007;0:373–80.

[25] Pollitt M, Nance K, Hay B, Dodge RC, Craiger P, Burke P, Marberry C, Brubaker B. Virtualization and digital forensics: a research and education agenda. J. Digit. Forensic Pract. 2008;2(2):62–73.

[26] Haeberlen A. A case for the accountable cloud. In LADIS'09: 3rd ACM SIGOPS International workshop on large scale distributed systems and middleware, 2009.

[27] SP 800-57, Part 1, Recommendation for Key Management: General, July 2012.

[28] SP 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, March 2007.

[29] SP 800-56B, Recommendation for Pair-Wise Key Establishment Using Integer Factorization Cryptography, August 2009.

[30] SP 800-56C, Recommendation for Key Derivation through Extraction-then-Expansion, November 2011.

[31] SP 800-132, Recommendation for Password-Based Key Derivation, Part 1: Storage Applications, December 2010.

[32] Prasad Saripalli, and Ben Walters, (2010). "QUIRC: A quantitative impact and risk assessment framework for cloud security". IEEE 3 International Conference on Cloud Computing.

[33] Verizon RISK Team, (2012). Data Breach Investigations Report (DBIR).

[34] Francisco Rocha, Thomas Gross, and Aad van Moorsel, (2012). "Defense in-depth Against Malicious Insiders in the Cloud". IEEE International Conference on Cloud Engineering.

[35] Peng Li, Debin Gao, and Michael K. Reiter, (2012)."Mitigating Access-Driven Timing Channels in Clouds nd using Stop Watch". The 42 Annual IEEE/IFIP International Conference on Dependable Systems and Networks.