Analysis of cloud computing attacks and countermeasures

Conference Paper · January 2016

DOI: 10.1109/ICACT.2016.7423295

Salam Khanji
Zayed University
14 PUBLICATIONS **45** CITATIONS
SEE PROFILE

Omar Alfandi
Zayed University | Georg-August-Universität Göttingen
97 PUBLICATIONS **400** CITATIONS
SEE PROFILE

Huwida E. Said
Zayed University
32 PUBLICATIONS **727** CITATIONS
SEE PROFILE

Analysis of Cloud Computing Attacks and Countermeasures

Raja Mohamed Jabir, Salam Ismail Rasheed Khanji, Liza Abdallah Ahmad, Omar Alfandi,Huwida Said
College of Technological Innovations, Zayed University, 144534, UAE
{M80006379, M80006416, M80006385, Omar.AlFandi, Huwida.Said}@zu.ac.ae

Abstract— **Business enterprises are competing to get their applications up and running faster with improved manageability and less maintenance by reassigning the ability to IT specialists to adjust resources in order to meet business fluctuating needs. This can be achieved through utilizing the Cloud Computing Model which is distinguished by 'pay as you go' model. The model offers solutions of storage, convenient and on-demand access to a shared pool of configurable computing resources. As with any novel technology, Cloud Computing is subject to security threats and vulnerabilities including network threats, information threats and underlying infrastructure threats. In this paper, we present a framework by which penetration testing is conducted to highlight possible vulnerabilities within our private Cloud Computing infrastructure, simulate attacks to exploit discovered vulnerabilities such as Denial of Service (DoS) and Man-in-the- Cloud attacks , apply countermeasures to prevent such attacks, and then to exemplify a recommended best practice protection mechanism which will contribute to the Cloud Computing security.**

***Keywords*— Cloud Computing; Man in the Cloud; DDOS; OpenStack.**

I. Introduction

The Cloud computing industry is experiencing a tremendous growth in usage. Along with that growth, there is an increase in security concerns. Cloud Computing faces traditional security concerns, in addition to security concerns that are unique to Cloud systems such as: computational out sourcing and resource sharing. Constantly, new attacks and threats are reported targeting the Cloud. Just recently Imperva, a leading provider of cyber and data security products, has warned its users of a new attack(Man-in-the-Cloud)[1]that targets Cloud applications.

This paper aims to understand the vulnerabilities and threats that Cloud systems are facing through a series of experiments on a private Cloud deployed using OpenStack, and simulating the DoS and Man-in-the-Cloud attacks on Cloud storage applications such as Dropbox. Moreover, it intends to create a framework in which its findings serve as guidelines toward exploring Cloud computing vulnerabilities by utilizing penetration testing techniques. Two testing modes, black box and white box, are deployed to further explore and evaluate security issues in a private Cloud that has been already built using Ubuntu OpenStack. Vulnerabilities are addressed, possible exploits and attacks are studied, and countermeasures are suggested accordingly. The paper will be organized as follows: Section II discusses related work. Section III presents the proposed methodology, while section IV illustrates the practical demonstration of the research work. Finally, section V discusses the results and findings of the experimental framework, and section VI presents a conclusion for the paper.

II. Literature Review

Cloud Computing has been capturing the interest of many organizations as well as academic entities due to its cost effectiveness and capabilities. Despite Cloud technology being adopted across the world, there is less literature on its security issues and the increase in Cloud attacks.

The diversity of Cloud computing models poses a security risk where different types of attacks are now targeting the Cloud infrastructure. Ali et al. [2] provided a broad explanation of Cloud infrastructure and the different available Cloud models by highlighting security threats targeting the communication layer, virtual network vulnerabilities, and users limited control on their data.

From a security standpoint, it is imperative to understand the security vectors which might threat or exploit possible vulnerabilities in the Cloud computing infrastructure. The main gaps are recognized by Khorshed et al.[3] as trust issues, security threats, security risk and some other specific Cloud computing related issues. Authors also suggested a proactive threat detection model which uses modern machine learning techniques to detect and classify an attack when it occurs, alert related parties (such as :system admin) about the attack type, take combating action, and generate information on the type of attack by examining the pattern.

The most predominant Cloud attacks are the Distributed Denial of Service (DDoS) or DoS attacks, and Man-in-the-Cloud attacks. The literature explored Cloud DoS attacks, where Choncka et al.[4] analyzed how HTTP DoS and XML DoS (H-DoS/X-DoS) attacks affectCloud systems using real attack traffic. They demonstrated how Cloud TraceBack, a service-oriented architectural, which can be used to identify the source of an attack within a short period of time. Moreover, they utilized the Cloud Protect, a back propagation neutral network, that was trained to detect and filter most of the H-DoS/X-DoS attacks.

Moreover, Darwish et al.[5] researched an original Cloud based authentication (CSA) protocol suite and demonstrated its effectiveness in protecting against external DoS attacks and against internal DoS vulnerabilities. It consists of three sets of protocols: the registration protocol, the CSA adaptive-based identification protocol, and the authentication protocol. This protocol allows the Cloud system to recognize genuine requests and pass them to the Cloud applications to be protected against internal and external DoS threats. However, the limitation of this protocol is that it only works in the SaaS layer of Cloud computing which makes it ideal to the private Cloud only. Other solutions were proposed to protect the Cloud infrastructure by utilizing existing technologies such as firewall and encryption schemes.

III. METHODOLOGY

Penetration testing is a technique to evaluate the security perimeters for the target of evaluation by simulating an attack to discover vulnerabilities that would be exploited by an attacker. The test includes a comprehensive analysis of the system configurations, designs, weaknesses, and technical flaws. From an administrator standpoint, it is crucial to check the level of system information disclosure as well as exploits which would make the system vulnerable for outside adversaries. Our methodology suggests a framework which consists of the following phases as illustrated in figure1.

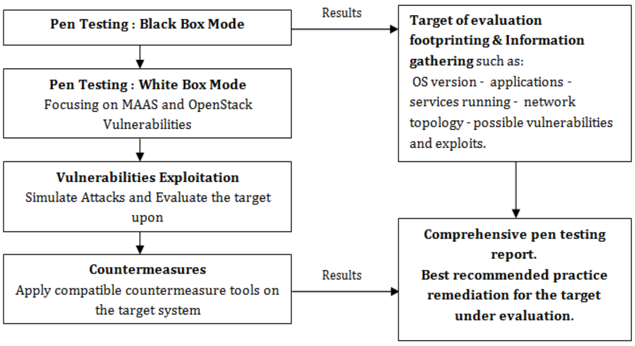

Pen Testing : Black Box Mode
Pen Testing : White Box Mode
Focusing on MAAS and OpenStack Vulnerabilities
Vulnerabilities Exploitation
Simulate Attacks and Evaluate the target upon
Countermeasures
Apply compatible countermeasure tools on the target system
Results
Results
Target of evaluation footprinting & Information gathering such as:
OS version - applications - services running - network topology - possible vulnerabilities and exploits.
Comprehensive pen testing report.
Best recommended practice remediation for the target under evaluation.

Fig.1 -Framework Phases

- *Phase One:* **Black Box Pen Testing**: We were assuming that we had no prior knowledge about the target of evaluation; we were neither aware that it is a Cloud environment, nor certain what technologies were being deployed. A comprehensive scan was conducted to probe as much information as possible about target-related vulnerabilities and exploits.

- *Phase Two:* **White Box Pen Testing**: In this mode of testing, we knew what type of environment we were testing and what specific vulnerabilities related to technologies used in deploying the Cloud as in OpenStack (Juju and Maas) were explored and evaluated.

- *Phase Three:* **Countermeasures Implementation**: Suggested countermeasures and best practice recommendations were proposed to protect our private Cloud environment from possible technology-specific vulnerabilities discovered in phase two.

IV. EXPERIMENTAL SETUP

Following are the experimental scenario and the proposed framework phases' specifications:

A. Experimental scenario

1)Cloud Infrastructure: The deployed infrastructure is made up of seven machines; one Ubuntu server 14.04.3LTS, and six Ubuntu Desktop14.04 nodes. All the seven machines are connected as depicted in figure2. Each of the seven machines has two hard drives capacity of 500GB and 250GB respectively, to offer Ceph redundancy. The Canonical Distribution's OpenStack Auto pilot for Ubuntu has been utilized to build, manage, and monitor the Cloud computing layer, by using tools; JuJu and MAAS Canonical tools [6,7].

2)Port and Network Scanning: We used Nmap which is a tool that is used for host discovery to determine alive hosts on the network along with details about it such as: OS, open ports and services.

3)Penetration Testing Tools: We mainly focused on using the Kali virtual machine along with Armitage vulnerability scanner.

Through Armitage, we would scan for exploits, launch attacks, and get exploit recommendations as well. Another tool that was used for scanning is the popular tool Nessus from Tenable. Nessus is a tool that can be used for vulnerability scanning and vulnerability management.

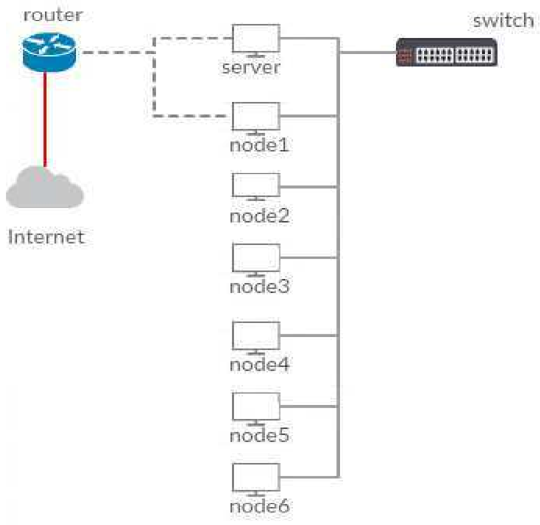


Fig.2 -Cloud Infrastructure[8]

B. Framework Phases

Phase One: Black Box Pen Testing

The first step was to perform host discovery through connecting to the network and launching an Nmap scan that was used to discover alive hosts on the network. Next, we used Nmap to scan the open ports of the active hosts detected in the previous step in addition to details about the services running and OS. The results of the Nmap scan are shown in table I. Next, after the IP address of the web server that is deployed on the private Cloud environment was identified, we conducted a vulnerability scan using the Armitage in Kali Linux virtual machine, as we were then familiar with the OS (Linux) and the open ports. We used attacks in Armitage that focused on Linux and the services found on the open ports. Table II summarizes the scanning findings of the Armitage. The attack module listed in table 2 are suggested exploits by Armitage, these attacks are known attacks for the available open services detected on the target and specific to the target operation system. Another scanning tool which was used to further cross-validate our findings was Nessus, Table III summarizes Nessus findings while report scan can be found in[9].

TABLE1. NMAP RESULTS

Open Ports	**Service/Version**
22/tcp	ssh/OpenSSH6.6.1p1Ubuntu2ubuntu2.3 (Ubuntu Linux)
25/tcp	smtp/Postfix smtpd
53/tcp	Domain
80/tcp	http/Apache httpd2.4.7(Ubuntu)
3128/tcp	http-proxy/Squidhttpproxy3.3.8
3260/tcp	Iscsi
5432/tcp	postgresql/PostgreSQLDB
7911/tcp	omapi/ISC(BIND|DHCPD)OMAPI
8000/tcp	http-proxy/Squidhttpproxy3.3.8
53/udp	Domain

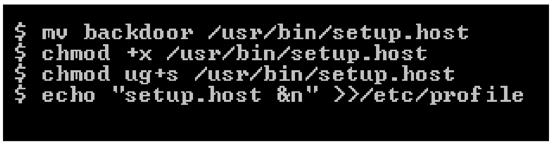
TABLE2. ARIMITAGE RESULTS

Attack Modules	**Affected port and service**
exploit/linux/ssh/f5_bigip_known_privkey	22/tcp/ssh
exploit/linux/http/…	80/tcp/http
exploit/linux/proxy/squid_ntlm_authenticate exploit/linux/samba/…	3128/tcp/http_proxy 8080/tcp/http_proxy
exploit/linux/postgres/…	5432/tcp/postgresql

TABLE3. NESSUSR ESULTS

Scan type	**Vulnerabilities**	**Affected port and service**
Advanced Scan& Basic Network Scan	1High 11Medium 3low	3260/tcp/isci-target 53/udp/dns, 8000/tcp/http_proxy, 25/tcp/smtp 22/tcp/ssh,25/tcp/smtp
Web Application Test	1Medium	80/tcp/www

Moreover, Armitage ssh_login module was used to test a range of ssh logins (from a custom dictionary) on the targeted machine to perform a brute force attack. The password was successfully cracked and it was possible to open a session with the target and use it to interact with the target and to create a backdoor on the target machine. We first created a Metasploit payload executable using the following module: payload/linux/x86/shell_reverse_tcp. Next, alistener was set up. Lastly, the backdoor was uploaded to the target machine as showing figure 3.



```
$ mv backdoor /usr/bin/setup.host
$ chmod +x /usr/bin/setup.host
$ chmod ug+s /usr/bin/setup.host
$ echo "setup.host &n" >>/etc/profile
```

Fig.3 – backdoor uploaded to target machine using Armitage

Phase Two: White Box Pen Testing

In this phase, we had a prior knowledge about the target of evaluation and we already knew it was a web server which manages a Cloud service. We knew how many nodes were connected and what operating systems were being deployed. Our goal was to address and test those vulnerabilities related to Cloud computing environment which causes threats to the environment performance. According to the report published by Cloud Security Alliance (CSA) [17], the top threats to Cloud computing are: Man-in--the-Cloud attack, Denial of Service attack, and attacks targeting OpenStack components.

- **Man-in-the-Cloud Attacks**

One of the most common attacks witnessed in 2015 is Man-in-the-Cloud attack, which is an attack that targets file storage/synchronization applications such as Dropbox and Google Drive. The attack depends on exploiting the applications' synchronization protocols and end-user authentication token. The attack is built on accessing a targeted victim account by authenticating as the victim without the need to crack their password; which hinders the detection process. Figure 3 in demonstrates the scenario. Cloud file storage and synchronization applications perform user authentication by implementing different mechanisms such as encryption. Google Drive application authenticates users by storing a device associated authentication key in the device registry with the value name 'OAuthToken_**'[1].

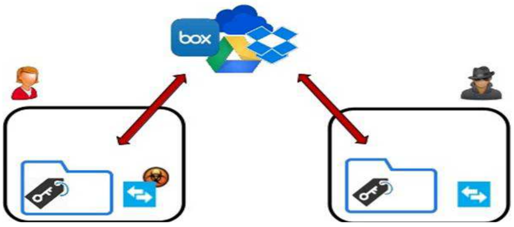


Fig.3 – Man-in-the-Cloud Attack Scenario [1]

Man in the Cloud attack comes in various forms such as:1) Account Sharing: This is when an attacker shares the victim's account, authenticating as the victim being able to access all the synchronized files and probably manipulating them. 2) Credential Swapping: The attacker swaps his authentication/synchronization token with the victim's tokens. As a result the victim is being misled to use the attacker account as if it were his. Authors simulated the Man in the Cloud by creating a script (script 1 in [18]) to replicate a victim Dropbox authentication file to the attacker machine.

- **Denial of Service (DoS) Attacks**

In our framework, we have focused on semantic DoS, where we flooded the web server with HTTP requests, and SYN requests as well. For flooding our target web server with HTTP requests, we have utilized the Low Orbit Ion Canon (LOIC) [10] which is an open source tool used to perform DoS on small servers by sending UDP, TCP, or HTTP requests to the victim server.

For the SYN flood; the attack exploits the TCP/IP three-way handshake to establish a connection. In the SYN flood attack, massive number of SYN packets are sent with random IP address to the destination server, and they are queued in the server's buffer to use both memory and server's resources for the purpose of causing it to crash or to hang. The code used was written in C language and launched from Kali Linux machine (Script 2 in [18]). It sends plenty SYN request to the IP address of the web server of the private Cloud, each time with a spoofed IP address. The impact of each attack on the private Cloud computing environment will be discussed in Section VI Results and Discussion.

- **OpenStack Components Attacks**

The main components identified as the core of OpenStack can be separately targeted by attackers due to existing vulnerabilities that can be exploited. Due to unperceived issues, the components couldn’t be deployed on the system and therefore none of the vulnerabilities could be verified or attacks could be tested. However, an example of documented cases are listed below.

1) Nova: Nova is the OpenStack compute written in python and it uses the fabric controller. It is considered the main part of the IaaS system as it manages and automates pools of computer resources[11]. Attackers can exploit Nova’s network configuration to reach the host on the same virtual network. Booted instances can be used to check the address of the gateway and then connect to the sshd service on the host system[12].

2) Horizon: Horizon is the canonical implementation of the OpenStack dashboard. It provides the user with an interface to access OpenStack services[11]. The attacker can exploit the default settings in horizon which uses the signed cookie to store the session state on the client side and steal the cookie using sniffing techniques or gaining access to the target system. The attacker can then use the stolen cookie to impersonate the target [13].

Phase Three: Countermeasures Implementation

Once penetration testing is conducted utilizing different testing modes, a set of countermeasures can be suggested as a mitigation plan in order to secure the target of evaluation. For our target, the diagram in Figure 4 suggests a complete plan to protect our private Cloud infrastructure from potential threats and vulnerabilities.

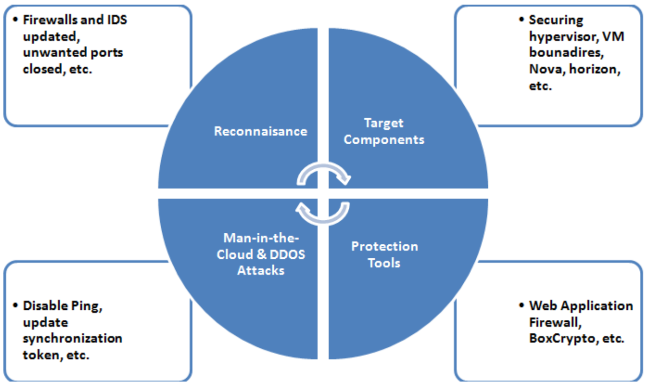

• Firewalls and IDS updated, unwanted ports closed, etc.
• Securing hypervisor, VM bounadires, Nova, horizon, etc.
Reconnaisance
Target Components
Man-in-the-Cloud & DDOS Attacks
Protection Tools
• Disable Ping, update synchronization token, etc.
• Web Application Firewall, BoxCrypto, etc.

Fig.4 – Suggested Countermeasures

Reconnaissance is the first step in which the attacker gathers information from all possible public resources. As illustrated in Phase one of the experimental framework; various reconnaissance techniques were utilized as in host detection, port scanning, and vulnerabilities scanning. Administrates need to secure their systems and networks against port scanning by following these measures:

1. Firewalls and IDS have to be updated and configured to detect and block probes sent from the attacker to perform the scan.
2. Sensitive information needs to be concealed.
3. Routing and Filtering mechanisms need to be configured in a way that they don’t allow the attacker to bypass using any source ports or source routing techniques
4. Unnecessary ports have to be closed using a custom rule set.

Moreover, each Cloud delivery model has its own security issues and threatening vectors. In our experimental framework, the IaaS delivery model was utilized for which we can summarize its security issues as follows:

1.***VM Security***: to secure the VM operating systems and workloads from the traditional threats as in viruses and worms. This is the responsibility of each tenant to use the convenient security perimeter which better suits his needs as in anti-virus, firewalls, etc.

2. ***Images Repositories Security***: to secure images which are the templates loaded from the image repositories; compromised image can be easily injected to this repositories which would maintain the owner information and to be used by a new tenant. This is the Cloud provider's responsibility.

3.***VM Network Security***: It is crucial to secure the virtual network infrastructure shared among multiple tenants which would be affected by IP protocols and DNS servers' vulnerabilities. This is considered part of the Cloud provider's responsibility.

4. ***VM Boundaries Security***: to secure server's CPU, memory, I/O, and NIC which sometimes are shared with other VM co-exists on the same physical server. This is again part of the provider's responsibilities.

5. ***Hypervisor Security***: to secure the orchestration layer which controls, monitors, and organizes the physical resources and maps them to virtualized resources.

Breaking the hypervisor would affect all the VMs since it would be traced and data are unencrypted. The service provider who provides this service (Canonical OpenStack in our model) is responsible to secure it by updating the patches and make them available to defeat any bug or vulnerable component.

In our experimental framework, we focused on tools which would contribute to securing our IaaS Cloud delivery model from different perspectives (as mentioned in the section above). These tools shall be integrated as part of the countermeasures suggested after a penetration tester conducts the test, summarizes the findings, and recommends best practice methodology to mitigate attacks such as those simulated. We have deployed two different tools as in below:

1. *Web Application Firewall(WAF) Testing Framework:* A free software tool which looks for web applications vulnerabilities, and generates both legitimate and attack traffic to evaluate if the Cloud environment can stop such traffic or even detect such malicious traffics [14]. This tool would help the Cloud provider to support protection of tenants' data and critical applications.
2. *BoxCryptor:* A free software which can be used by the Cloud tenants who wish to protect their information utilizing encryption. This application is supported by different digital medium as in PC running Windows and Mac operating systems, as well as mobile phones; Android and IOS. This protection mechanism protects the tenant's side only [15].

Finally, core components of OpenStack have to be constantly updated and secured to avoid exploiting any vulnerabilities such as the ones mentioned earlier in phase two. Following are examples of the mitigations to secure the mentioned vulnerabilities.

A vulnerability exists in Nova that can allow guest VMs to connect to host services. This can be solved by preventing the critical services from binding to the nova controlled interfaces. To illustrate using an example, sshd service default settings should be changed. Instead of allowing binding to all interfaces and all local addresses, the service should be configured to a specific interface and address[12]. Another vulnerability mentioned is the session fixation vulnerability in the Horizon component of OpenStack. To avoid this vulnerability, horizon has to be configured to use different session backend such as: memcache session instead of signed cookies and always enable SSL for Horizon[13].

V. RESULTS AND DISCUSSION

The IaaS cloud model suggests the dependency of the stacked components of Cloud which further complicates the security model to be utilized. Each layer's security depends on the security of all beneath layers. However, each layer differs in security controls since each has its own security requirements and vulnerabilities. Any breach in security in any of the stacked layer would indeed impact the security of the whole Cloud environment.

Assessments were performed first using Nmap scanner and then using Nessus and Armitage. Nmap scanner results revealed the active nodes in the network in addition to details about the OS, open ports and the running services. The information retrieved from the IP scanning and port scanning helped us in determining what hosts we need to scan for vulnerabilities. Using Nessus and Armitage, we scanned the server and uncovered various vulnerabilities linked to the open ports. Studying the vulnerabilities listed in both reports can easily help attackers in exploiting them.

Nessus classifies detected vulnerabilities according to their risk severities by using the vulnerability score assigned by the Common Vulnerability Scoring System (CVSS).For example, the Nessus scanner has detected a medium level vulnerability stating that the SSL certificate of the server can't be trusted. This vulnerability can be exploited by an attacker to carry out Man-in-the-Cloud attacks against the remote host. Another medium level vulnerability revealed by Nessus is the HTTP Proxy Post request relaying which might enable the attacker to go through the firewall, by connecting to sensitive ports like port 23 using the proxy. Moreover, the Armitage vulnerability scanner revealed various vulnerabilities. However, we weren't successful in compromising the target when attempting to manually or automatically exploit those vulnerabilities using the Armitage's attacks module. Therefore, the brute force attack using the ssh_login module was successful in gaining access to the target and opening a session that was used to interact and upload a backdoor to maintain access to the target.

As depicted in Phase 2 of the experimental framework, we simulated the Man-in-the-Cloud attack on Dropbox authentication file. Our experiment results illustrated that attackers can easily transfer user authentication files into another device; which can lead to unauthorized remote access, identify fraud, steal confidential information or even malware implementation. Man-in-the-Cloud compromises a large number of accounts due to these accounts popularity, not only for personal use but also in large organizations. This attack can compromise Cloud applications without requiring users' credentials which adds to the difficulty of detection; as it does not require executing any malicious codes on the client side, on the contrast it can be performed through basic means such as users clicking on a phishing email link or social engineering.

The second simulated attack in Phase Tow is DDoS attack which would have negative impact on the Cloud environment. Resources as in network bandwidth and storages will be unavailable to legitimate users due to the high consumption both HTTP-DoS and SYN flood caused. In a similar paper [16] where the DoS traffic affecting CPU usage was studied using a setup similar to the setup discussed in this project, Under a 10 Mb/s TCP DDoS, the hypervisor KVM consumed the CPU time of an entire core merely delivering SYN packets to the VM and returning SYN-ACKs to the network. When the attack rate was increased to 100 Mb/s, the KVM hypervisor consumed almost half of the systems total CPU resources while processing the attack traffic. If small or medium- sized companies are utilizing all their resources as well as their business applications and services on Clouds, unavailability will cause customers dissatisfactions and even worse will cause business loss if mitigation plans are not initiated. Due to the IaaS's specifications, this model imposes threats which can expose vulnerabilities to be exploited by attackers. For instance; the virtual networks which are shared among multiple tenants in the Cloud environments would increase the possibility to exploit vulnerabilities in the IP protocols as depicted in the simulated DoS attacks. Moreover, the virtual machines co-existing on the same physical server would share memory, CPU, I/O and NIC. Securing this VM boundaries is the responsibility of the Cloud provider.

As illustrated from Phase Three of the experimental framework; there are plenty of free software and protection solutions which can be used to protect critical assets by deploying the Cloud computing environment. Some tool scan provide a protection level while monitoring and supervising the use of the shared pool of resources. These techniques enable Cloud administrators to monitor the network traffic, vulnerable VMs, vulnerabilities in services and applications offered in order to provide ultimate protection which is the main goal for any Cloud provider. Available tools can be utilized to encrypt data being synchronized, stored or processed.

The followings are best practice recommendations as countermeasures when considering Cloud Computing security:

1. File Synchronization vendors should implement account access restriction to a specific set of devices identified by account owner.
2. Users should be notified for unusual account access, by sending alert emails or messages.
3. Users should explicitly revoke all tokens periodically.
4. Enterprises should enhance security by using a controlled Cloud access solutions; this solution intends to monitor usage and access of Cloud services.

VI. CONCLUSION

Cloud Computing is the topic of the era as different sectors organizations, regardless of their size, are adopting it. This is due to its cost effectiveness and optimized resource utilization. In this paper, the Authors examined the current state of Cloud security, by performing penetration testing to assess the security of Cloud environment and its underlying components. Two penetration testing modes have been utilized, Black box assuming no prior knowledge of the system under evaluation in which various scanning activities were conducted to determine the hosts IP address, operating system, and open ports. The second mode in the proposed framework is White box testing, in which Authors were aware that the system under evaluation is a Cloud environment; in which penetration attacks were simulated to target specific Cloud vulnerabilities. The two attacks performed were Distributed-Denial-of-Service and Man-in-the-Cloud attacks. Both attacks simulated the vulnerabilities of Cloud to risks such as data breach, indentify fraud, compromising integrity, availability and confidentiality. Despite Cloud introduced values, organizations should conduct proper assessment to its associated risks to select the suitable Cloud infrastructure. This research highlighted the vulnerabilities of Cloud and emphasized on the need for implementing enhanced security measures.

REFERENCES

[1] Imperva,'Man in the Cloud (MITC) Attacks', 2015. Retrieved from: https://www.imperva.com/docs/HII_Man_In_The_Cloud_Attacks.pdf . [Accessed 11/12/15 12:32].

[2] Ali, M., Khan, S. U., & Vasilakos, A. V. (2015). Security in Cloud computing: Opportunities and challenges. Information Sciences, 305, 357-383

[3] M. Khorshed, A. Ali and S. Wasimi, 'A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in Cloud computing', Future Generation Computer Systems, vol. 28, no.6, pp. 833-851, 2012.

[4] A. Chonka, Y. Xiang, W. Zhou and A. Bonti, 'Cloud security defence to protect Cloud computing against HTTP-DoS and XML-DoS attacks', Journal of Network and Computer Applications, vol. 34, no. 4, pp.1097-1107, 2011.

[5] M. Darwish, A. Ouda and L. Capretz, 'A Cloud-based secure authentication (CSA) protocol suite for defense against Denial of

Service (DoS) attacks', Journal of Information Security and Applications, vol. 20, pp. 90-98, 2015.

[6] F. Stroud. (2015) Metal-as-a-service (maas). [Online]. Available:http://www.webopedia.com/TERM/M/metal-as-a-service maas.html.

[7] F. Stroud. (2015) Juju. [Online]. Available: http://www.webopedia.com/TERM/J/juju.html

[8] Masri R. & Hassan H. (2015). ' Virtualization Security in Cloud: Installation, Configuration, and Testing using OpenStack'. Retrieved From:https://www.dropbox.com/s/kxrsu5qsrvse1ts/virtualization%20security%20in%20cloud.pdf?dl=0. [Accessed 11/1/15 7:30].

[9] https://www.dropbox.com/sh/ip6n3v4cy7bknh8/AAA_Df1YbD2A-5jX5Of1wW0ta?dl=0

[10] Low Orbit Ion Canon (LOIC). Retrieved From: http://resources.infosecinstitute.com/dos-attacks-free-dos-attackingtools/ [Accessed 11/11/15 7:31].

[11] 'OpenStack Docs: Developers'. [Online]. Available: http://docs.openstack.org/developer/openstack-projects.html. [Accessed:13- Nov- 2015].

[12] N. Kinder, 'OpenStack Open Source Cloud Computing Software Nova Network configuration allows guest VMs to connect to host services', openstack Cloud software, 2015. [Online]. Available: http://lists.openstack.org/pipermail/openstack-dev/2014-June/038664.html. [Accessed: 13- Nov- 2015].

[13] Lists.openstack.org, 'OpenStack Open Source Cloud Computing Software Session-fixation vulnerability in Horizon when using the default signed cookie sessions', 2014. [Online]. Available: http://lists.openstack.org/pipermail/openstack/2014-June/007990.html. [Accessed: 13- Nov- 2015].

[14] Waf Testing Framework. Imperva Inc. Retrieved from: https://www.imperva.com/lg/lgw.asp?pid=483. [Accessed 11/12/15 6:58].

[15] BoxCryptor. Retrieve From: http://www.boxcryptor.com/en/download. [Accessed 11/13/15 7:35].

[16] R. Shea and J. Liu, 'Performance of Virtual Machines Under Networked Denial of Service Attacks: Experiments and Analysis', IEEE Systems Journal, vol. 7, no. 2, pp. 335-345, 2013.

[17] Babcock C. (2014) 9 Worst Cloud Security Issues. InformationWeek. Retrieved from http://www.informationweek.com/Cloud/infrastructureas-a-service/9-worst-Cloud-security-threats/d/d-id/1114085. [Accessd 11/7/15 20:50].

[18] Jabir R., Khanji S., Ahmad L. (2015). Technical Documents. Retrieved from: https://www.dropbox.com/sh/z04x2u94gch92jq/AADIRr_kOul012-BkB0Dfzy-a?dl=0

Raja Jabir Master student at the College of Technological Innovation at Zayed University, Abu Dhabi, UAE. She received her Bachelor of Computer Science from Abu Dhabi University, UAE in 2010.

Salam Khanji Master student at the College of Technological Innovation at Zayed University, Abu Dhabi, UAE. She received her Bachelor of Computer Science from University of Jordan, Jordan in 2003.

Liza Ahmad student at the College of Technological Innovation at Zayed University, Abu Dhabi, UAE. She received her Bachelor of Computer Science from the American University of Sharjah, UAE in 2009.

Omar AlFandi Associate Professor at the College of Technological Innovation at Zayed University. He holds a Doctoral degree (Dr. rer. nat.) in Computer Science and Telematics from the Georg-August-University of Goettingen - Germany in 2009. He received his M.Sc. degree in Telecommunication Engineering in 2005 from the University of Technology Kaiserslautern - Germany. Between 2009 and 2011, he enjoyed a Post-doctoral Fellowship at Telematics Research Group and he founded a Research and Education Sensor Lab where he is currently as Lab Advisor. Before that he carried his Doctoral Research as part of an Industry, Academia and Research centers collaboration European Union (EU) project.

Huwida Said Associate Professor in the College of Technological Innovation at Zayed University, Dubai, UAE. She received her Bachelor of Engineering (B.Eng.) in Electrical and Electronics Engineering from the University of Wales, Swansea, UK, and a Ph.D. in Computer Sciences from the University Of Reading, UK. In 2008, she received a National Science Foundation (NSF) fellowship grant to conduct a post-doctorate certificate in Information Assurance from the University of Maryland University College UMUC, Maryland USA.