Security Architecture based on Defense in Depth for Cloud Computing Environment

Theodoros Mavroeidakos
Department of Informatics
University of Piraeus
80, Karaoli & Dimitriou St.
GR-185 34, Piraeus, Greece
Email:
mavroeidakos.theodoros@gmail.com

Angelos Michalas
Department of Informatics and
Computer Technology Technological
Educational Institute of Western
Macedonia
52100, Kastoria, Greece
Email: amichalas@kastoria.teiwm.gr

Dimitrios D. Vergados
Department of Informatics
University of Piraeus
80, Karaoli & Dimitriou St.
GR-185 34, Piraeus, Greece
Email: vergados@unipi.gr

Abstract — **Cloud Computing constitutes an emerging computing paradigm consisting of elements of grid computing, utility computing and software-defined networks. The aggregation of these technologies offers a new environment for the deployment of services. Cloud computing environment provides capabilities which are unique covering the existing and future needs of organizations and companies. Moreover, this environment supports big data applications usually forming the core elements of research projects. Therefore cloud computing technology and big data are linked to each other. However, the capabilities of cloud computing environment create challenges concerning the security of data applications and its systems. In this respect, security issues are present on big data applications. By adopting the cloud computing environment, the provider has to incorporate security systems and policies in its infrastructure in order to mitigate the security threats. In this paper, multilayered security architecture is defined based on defense in depth. In this architecture the cloud infrastructure is divided into defense zones to achieve better security control. Additionally, intrusion detection system (IDS), honeypots and firewalls are incorporated alongside the defense mechanisms of the cloud infrastructure. In this way, a secure architecture is applied in which the end service is provided uninterrupted, while control over the level of security is maintained.**

Keywords - Cloud Computing, Defense in depth, Security architecture, Intrusion detection system (IDS), Honeypots, Firewalls, Big data applications

I. INTRODUCTION

The security architecture design for the network topology of cloud computing environment is a critical task, because of the factors that should be taken into consideration. The security of this computing paradigm is affected by the deployment models, the service models and the type of the end service. A promising deployment model achieving scalability, a well as, a satisfactory level of security at the same time is the hybrid one. Furthermore, there are three service models, the SaaS, the PaaS and the IaaS. There is a strong dependency among the service models concerning the security threats. Thus, in terms of the service model, the organization or company should migrate into IaaS, in order to mitigate security threats in the lower abstraction level. The end service could be a web application, dynamic web pages or a big data application such as Apache's Hadoop. Either way, the security components of the network architecture should provide protection from attacks which target the application layer of the OSI model. Moreover, cloud computing attacks which target data and information in general could not be eliminated however they can be faced. This task is entrusted in the security architecture, which is created by four principles defined in [1] the deterrence, the detection, the delay and the denial. To this end, the security architecture proposed in this paper consists of different types of firewalls creating cooperative defense zones. Furthermore, a honeynet is incorporated in order to attract the attackers, as well as, a distributed intrusion detection system capturing and analyzing the ingress and egress traffic of defense zones. Beyond these security entities, certain defense mechanisms of cloud computing such as security groups, virtualization and availability zones are orchestrated. The main objective of this architecture is to secure the operations of the cloud computing environment by collaboration among entities, without affecting cloud's capabilities.

The paper is organized as follows. In Section 2 the related research literature is reviewed while Section 3 contains the description of a cloud computing environment in which the security architecture was mapped. Section 4 contains the proposed security architecture. Section 5 contains information about the configuration of security systems. Section 6 contains information about the mitigation of threats and the evaluation in general of the proposed architecture while conclusions are drawn in section 7.

II. BACKGROUND

To address the issue of securing the cloud environment, security models and methods should be considered. In [2], a security model for cloud computing is proposed which incorporates OTP authentication, hashing algorithms, an encryption algorithm and a mechanism for recovery of data. Moreover in [3], four security models are summarized namely the Multiple-Tenancy Model of National Institute of Standards and Technology(NIST), the Risk Accumulation Model of Cloud Security Alliance(CSA), the Cube Model of Jerico Forum and the Security and Compliance Mapping Model. In particular, the cloud multiple-tenancy model of NIST is based on virtualization in order to satisfy security concerns. The approach of cloud risk accumulation model of CSA is based on the analysis of security dependencies of each service model in the infrastructure of the cloud provider. According to Jerico forum's cloud cube model, every service and deployment model should be classified with a security definition so that the appropriate actions take place against threats. In the mapping model of cloud, security and compliance contribute assisting the cloud provider to determine whether to accept or to refuse the security risks of the environment. Apart from the above models which approach the security of the cloud environment on the organizational level, the IDS Snort, can be used within the software defined networks of OpenStack as suggested on [4]. Also in [5], the intrusion responsive autonomic system is incorporated in cloud computing to manage the capacity of logs and execute detection of attackers orchestrating a big data environment for the analysis of logs. On the other hand, our proposed security architecture, describes methodologies aiming to secure the cloud computing environment on the physical level apart from the organizational approach. Furthermore it covers a wider spectrum of attacks, than those addressed in [4] and [5]. Moreover, the proposed approach is ideal for an organization or company which migrate its production infrastructure to cloud computing environment.

III. Cloud Computing Environment

The presented security architecture in Fig. 1 is mapped in the network topology of the cloud computing environment of OpenStack [6]. This environment consists of five interacting computing entities used for the following purposes:

- Controller nodes are responsible for the configuration of the environment and the orchestration of tasks in the other nodes.
- Compute nodes are the core component of the Infrastructure-as-a Service and house the instances which deliver the end service.
- Network nodes are points of control for the virtual networks of this environment and their main task is the networking of the instances.
- Block nodes offer space for the permanent storage of data and information. Volumes of this type of storage can be dynamically attached to instances.
- Object nodes are used for the storage of the cloud images which are necessary for the creation of instances.

The nodes of the environment are interconnected through the management network deploying OpenStack Services running Ubuntu 14.04. For the delivery of the end services, instances are created using, a dedicated OpenStack flavor and an image.

Furthermore, VLANs and a tunneling network are used between the network and the compute nodes. The instances used to deliver the end service, are grouped into independent availability zones and are orchestrated by the loadbalancers, the web application firewalls (WAFs) and the OpenStack instances. In each subgroup of computing entities on the availability zones, a security group is created according to which the network traffic is controlled to correspond only to legitimate ports.

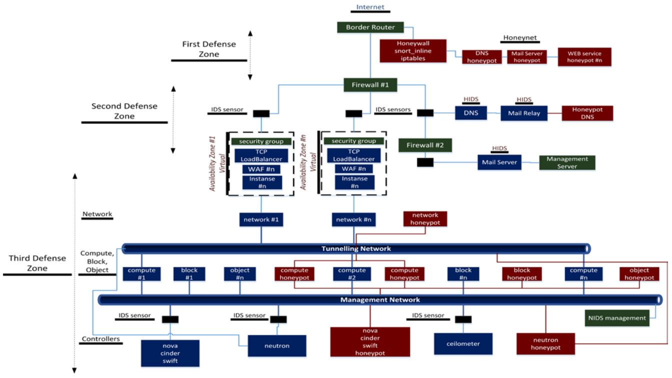

Fig. 1. Security Architecture.

IV. Proposed Security Architecture

According to [7], the main objective of organizations and companies is to ensure the confidentiality, integrity and availability of data and supporting systems. To construct the security architecture, the threats and vulnerabilities of the underlying infrastructure should be considered. Thus the proposed security architecture is defined by a set of distinct functional layers namely the perimeter defense, the deceptive, the detection and the cryptography. The collaboration diagram on Fig. 2 shows the priority among the layers of the architecture. By adopting the proposed security architecture based on defense in depth, it is necessary to create defense zones in order to classify the type of data in each and suitably protect them. Due to the network topology of the cloud computing environment, the users should access every zone of the infrastructure in order to use the end service. Accordingly, the security mechanisms proposed in each layer, are implemented in the defense zones.

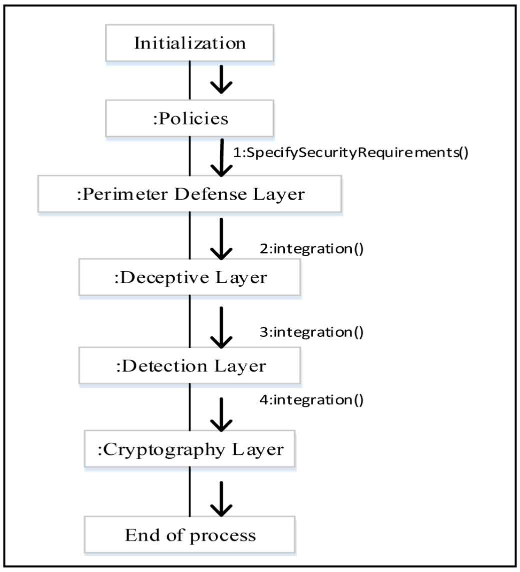

Fig. 2. Collaboration diagram of functional layers.

Apart from the security mechanisms of each zone, in order to cover the security needs of a cloud computing environment adequately, it is crucial to define a sequence of policies. According to them, the installation and configuration of security systems are performed. The policies should maintain the balance between productivity, functionality and security. Moreover, they should define the security responsibilities of the human factor and the effect of it on security systems. For this reason, an appropriate policy should be created in order to define the expected behavior of employees in relation to their authorization during their interaction with the environment. The policies have to set the necessary guidelines to encounter attacks in case of detection and the required response time. Table I shows the potential security systems that could be used in each layer.

TABLE I. POTENTIAL SECURITY SYSTEMS IN EACH LAYER

Layers	**Security systems**
Perimeter Defense	OSSEC[11], ModSecurity[12], OpenStack Security Groups
Deceptive	Second Generation Honeynet [8], Honeyd[13]
Detection	Snort [9]
Cryptography	Elliptic Curve Cryptography [10]

A. *Perimeter Defense Layer*

The main objective of the security mechanisms in this layer is the perimeter defense between the provided service and the rest of the Internet as well as between the systems of the defense zones. This layer provides the core security functionality, orchestrating security components to protect the classified data in defense zones without affecting the capabilities of the cloud computing environment. It consists of a border router, two stateful inspection firewalls, the security group of instances and the virtual WAFs as it is shown on Fig. 2. The border router performs packet filtering and constitutes the first defense mechanism of the environment. Subsequently, the first stateful inspection firewall is emplaced in order to create the first defense zone where a honeynet operates. The second defense zone extends among the first firewall, the security group of instances and the second firewall. Within the second defense zone, the DNS server and the Mail Relay operate. Moreover, the security group acts as a virtual firewall and its rules control the inbound and outbound traffic which reaches the autoscaling groups and their entities. The third defense zone lays behind the second firewall and the security groups. In this zone are located, the Mail server, the IDS management server as well as the production networks supporting the end services. The WAFs are emplaced between, the loadbalancers and the instances, in order to monitor information concerning protocols in the application layer. Additionally, certain servers on the second and third defense zone, such as the DNS, the Mail Relay and the Mail server, operate under the control of a host-based intrusion detection system.

B. *Deceptive Layer*

In this layer reside the deceptive systems which operate in every defense zone of the infrastructure. Accordingly, a honeynet is emplaced in the first defense zone between the external router and the first firewall in order to lure the attackers and detain them enough to detect them. Based on [8], the honeynet consists of the honeywall and high interaction honeypots. The honeynet can be used by the organization to identify new vectors of attacks and vulnerabilities of their systems. Furthermore, a number of honeypots are set up in key points of the second defense zone to detect attackers who bypass the first zone. These are low interaction honeypots which emulate legitimate services. Additionally, in the third defense zone a deceptive network of honeypots is created emulating the operations of the controller, the compute and the network nodes. These are honeypots of high interaction and clones of the production nodes, constituting a defense mechanism against insider threats. Under normal conditions, any ingress or egress traffic captured in honeypots should be considered malicious. Thus these systems facilitate notifications to security administrators concerning malicious events and delay the progress of an attack.

C. *Detection Layer*

The intrusion detection systems (IDSs) analyze the network traffic with a predefined ruleset identifying attempts of intrusion or attacks. The IDS adopted in our architecture is the open source tool, snort [9] consisting of three elements, the remote sensors, the management server and the monitor machine. In this security architecture, remote sensors are placed after the border router, the first firewall and the second firewall capturing the whole network traffic so as to increase the accuracy of detection and decrease the false positives. The management server is placed in a secure location, behind the second firewall in a different network by the production networks of OpenStack. However, in this placement, security threats continue to exist mostly because of insider threats. Furthermore, the monitor machine operates inside the management server so as the security administrator can manage the intrusion detection system locally mitigating security risks. At this point it has to be noted that the efficiency of detection is strictly connected with the ruleset of the remote sensors. For this reason, it is necessary not only to configure them correctly but also to update them continuously with new signatures of attacks.

D. *Cryptography Layer*

By this layer cryptographic methodologies are incorporated into the cloud environment such as the elliptic curve cryptography proposed in [10]. The procedures and function of cryptography should not contradict with the other layers. In addition, in case the encrypted data harden the operation of intrusion detection systems, then their emplacement is avoided. This happens because the detection of an intrusion is more important than the concealment of data. By detecting the intrusion the proper actions can be taken in order to mitigate the threat. Otherwise, in case of modification of the cryptographic system by the attacker, the data are made unusable. Additionally, the TLS protocol should be used so as to achieve end-to-end encryption between the clients and the web server which will provide the end service.

V. Configuration of Security Systems

The ruleset of the firewalls, the intrusion detection systems and the security groups are strictly connected to the provided level of security. The objective of each ruleset is to control the malicious inbound and outbound traffic in order to mitigate threats which target the data of the infrastructure and the systems of the Internet.

Accordingly, the border router executes packet filtering and constitutes the first point of inspection for the bidirectional network traffic. The configuration of the system should be performed according to [14], [15] and [16]. In this way, the ICMP traffic should be blocked entirely in order to avoid attacks against the TCP protocol such as blind connection-reset, blind throughput-reduction and blind performance-degrading attacks as well as UDP port scans. Moreover, the authorized network traffic concerns the servers on the second defense zone as well as the A records of DNS containing the ip addresses of loadbalancers. In this way the DNS server performs load distribution and the loadbalancers perform balancing of the network traffic load. Furthermore, the ip source routing feature of the border router should be disabled and the first fragment of a packet should contain a default quantity of information about the transport header. If the configuration is not feasible on the border router then it should be accomplished by the first firewall.

The security policy of the first firewall should contain customizations about embryonic connections, performing session lookups, checking the TCP sequence numbers and verifying the IP checksum. Moreover, the first firewall should behave as a redundant system in case of failure of the border router and as the main defense in case an attack overtakes it. Thus, the first firewall mitigates the same threats as the border router by adopting most of its configuration choices. Furthermore, firewall rules are configured concerning the allowance of IDS traffic for the communication of sensors with the management server. Additionally, the netblocks of IP addresses defined in the DROP and EDROP lists should be blocked due to the fact that they are hijacked and are involved in cyber-crime operations [17]. Moreover, the rules should be configured in such a manner [18] to avoid degradation of the end service. To this end the rules should be defined in the following order: firstly the deniability rules, secondly the allowance rules and finally the rules concerning general decisions. Table II describes indicative rules of the proposed configuration. Because of the nature of the environment, it is crucial to avoid negative effects on scalability and availability by using a firewall that supports a great number of concurrent TCP connections. In the opposite case the firewall will be the bottleneck of the infrastructure.

Accordingly, the second firewall protects the Mail server and the management server of the intrusion detection system. The configuration settings of this firewall do not differ from the first one but consist of a smaller number of rules. The administration practice used to configure the rules of this firewall is whitelisting because of the fact that the network traffic is finite between specified ip addresses. Such a practice is much safer than blacklisting due to the fact that there is an increasing number of vulnerabilities and attack vectors. Additionally, the second firewall should be the product of a different vendor than the first one in order to avoid exploitation of common vulnerabilities.

TABLE II. Indicative Firewall Rules

Rule	Direction	Protocol	SourceIP	SourcePort	DestIP	DestPort	Action
1	IN	TCP	172.16/12	Any	Any	Any	Block
2	IN	TCP	192.168/16	Any	Any	Any	Block
3	IN	TCP	DROPlist_ips	Any	Any	Any	Block
4	IN	TCP	EDROPlist_ips	Any	Any	Any	Block
5	IN	TCP	127.0.0.1	Any	Any	Any	Block
6	IN	TCP	Any	Any	LB_ip	80	Allow
7	IN	TCP	Any	Any	LB_ip	443	Allow
8	OUT	TCP	LB_ip	Any	Any	Any	Allow
9	OUT	TCP	LB_ip	Any	Any	Any	Allow
10	IN	ICMP	Any	Any	Any	Any	Block
11	OUT	ICMP	Any	Any	Any	Any	Block
12	IN	TCP	Any	Any	Any	Any	Block
13	OUT	TCP	Any	Any	Any	Any	Block
14	IN	UDP	Any	Any	Any	Any	Block
15	OUT	UDP	Any	Any	Any	Any	Block

The ruleset of every IDS sensor should be configured to detect patterns of attacks based on signatures. Attack signatures are used in Snort rules in order to identify attacks. These signatures should be updated because of the fact that new types of attacks are continuously created. The signatures of attacks are usually present either in the header part of a packet or in the payload. The security administrators of snort use perl compatible regular expressions to create signatures for attacks. Table III shows two indicative signatures that are used for the identification of cross-site scripting and SQL injection attacks. Additionally, Table IV shows indicative snort rules that can be used in order to mitigate the stealth scan of Nmap, Xmas, LOIC UDP DoS attack and LOIC HTTP DoS attack. Apart from this, the computing systems which perform the task of IDS sensors, have to be very powerful in order to avoid adding extra delay overhead.

TABLE III. Indicative Snort Regex Signatures

<table>
<tr><th rowspan="2">System</th><th colspan="2">Signatures and Attacks</th></tr>
<tr><th><i>Regex signatures</i></th><th><i>Attacks</i></th></tr>
<tr><td rowspan="2">Snort</td><td>/((\%3C)|<)((\%2F)|\/)*[a-z0-9\%]+((\%3E)|>)/ix</td><td>cross-site scripting</td></tr>
<tr><td>/\w*((\%27)|(\'))((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix</td><td>SQL injection</td></tr>
</table>

The dynamic detection technique of snort should be followed in order to identify attacks in real time. Apart from these systems, the WAFs are configured using the ModSecurity Core Rule Set (CRS) [19] following the network-based deployment. According to [20], this security mechanism can mitigate the XSS attack which targets the session ID of clients with the *httpOnly* flag. Furthermore it has the ability to flag the session cookies as secure avoiding thus to compromise the client's session cookie. Moreover, it supports content injection into http responses leading to in-browser inspection capabilities. The configuration choices presented, will be supplemented by the assignment of an ideal number of instances in each WAF avoiding delays. Lastly, the security administrators should possess thoroughly security knowledge to avoid security misconfigurations which create attack vectors.

VI. Security Architecture Evaluation

By adopting the proposed security architecture, mitigation of threats is achieved. However, threats and vectors of attacks continue to exist. For this reason, it is crucial to update and monitor the security systems locally and continuously. Table V presents the most dangerous attacks and the systems which restrict them. Certain types of attacks such as DoS, DDoS and flood attacks are strictly connected to embryonic connections and ip spoofing [22] [23]. In this way the border router, the firewalls and the security groups have the ability to mitigate them. Attacks which target the transport layer of OSI model can be treated by the systems of the perimeter defense layer. According to [22], WAFs provide HTTP protection against HTTP DoS attacks, common web attacks such as cross-site scripting and SQL injection. Moreover, WAFs offer Trojan protection, webshell detection and anti-virus scanning of file attachments. Furthermore, the IDS with adequate optimization should protect the infrastructure [30] against several types of attacks including buffer overflow attacks, stealth port scans, smb probes, cgi attacks, os fingerprinting, types of vulnerability scans, viruses, worms and DDoS attacks based on worms. The honeynet tracks the moves of the attacker and gathers forensic information about attacks. Finally, according to [8], the honeynet has the ability to identify new vectors of attacks, malicious behavior and 0-day exploits.

TABLE IV. Indicative Snort Rules

System	Rules and Attacks	
	Rules	***Attacks***
Snort	alert tcp $EXTERNAL_NET any -> $AVAILABILITY_ZONE_#n any (msg:"Xmas Scan -sX"; flags:FPU,12; ack:0; window:2048; threshold: type both, track by_dst, count 1, seconds 60; classtype:attempted-recon;)	Nmap Xmas Scan
	alert udp $EXTERNAL_NET any -> $AVAILABILITY_ZONE_#n any (msg:"LOIC UDP flooding"; threshold: type threshold, track by_src, count 100, seconds 5;)	LOIC UDP DoS
	alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SLR - LOIC DoS Tool (HTTP Mode)";flow: established,to_server; content:"|47 45 54 20 20 48 54 54 50 2f 31 2e 30 0d 0a 0d 0a 0d 0a|"; threshold: type threshold, track by_src, count 10 , seconds 10;)[21]	LOIC HTTP DoS

TABLE V. Attacks and security systems

Attacks	Security systems					
	Border Router	*Firewall #n*	*Honeypots*	*IDS*	*Security Group*	*WAF*
Intranets IP address spoofing	✓	✓			✓	
Tiny fragment attacks		✓				
Buffer Overflows			✓	✓		
Port scans	✓	✓	✓	✓	✓	
OS fingerprinting			✓	✓	✓	
Web attacks				✓		✓
Trojan attacks						✓
Viruses and Worms				✓		
Insider Threat			✓	✓		
Attacks on virtualization						✓
DoS and DDoS attacks		✓		✓		
HTTP DoS and DDoS				✓		✓
0-days exploits			✓			

Based on the above considerations, it is evident that the security systems of the architecture protect the infrastructure against specified attacks and supplement each other.

The evaluation of the presented architecture is performed by automated tools. By using the DDOSim[25], R-U-Dead-Yet (RUDY)[26], LOIC[27], Nmap[28], Nessus[29] and Tcpdump[30] tools, it was feasible to assess the outcome in case of real-time attacks. The implementation of stealth scans with Nmap and Nessus on the second and third defense zones, were identified by the IDS sensors. Moreover, a DoS attack performed from the external network to the ip address of a LoadBalancer by sending TCP requests using LOIC was identified and monitored by the IDS sensor of the availability zone hosting the loadbalancer. The WAF identified an HTTP DoS attack with valid requests performed by DDOSIM as well as an HTTP DoS attack performed by RUDY. During the evaluation, Tcpdump was used in order to monitor the network traffic associated with the attacks. Table VI shows the average duration of attacks or scans as well as the time required by the security systems to identify the threat and alert the administrators. It has to be noted that the response time of the security systems is associated with the magnitude of the ruleset and the network throughput. In this case the network throughput as well as the cpu load of nodes remained constant throughout these tests.

TABLE VI. Response Time to Attacks and Scans

	Security Systems	
Attacks	***IDS Alarm***	***WAF Alarm***
Nessus Advanced Scan	1,6min./10sec.	
Nessus Web App Tests	29min./1,1min.	
Nmap SYN Stealth Scan	13,22sec./5,3sec.	
Nmap Xmas Scan	14,38sec./5.4sec.	
Nmap OS fingerprinting	14,93sec./4,9sec.	
Nmap NSE Shellshock Script	13,48sec./5,3sec.	
DDoSIM HTTP DoS Attack		25min./30sec.
RUDY DoS Attack		22min./44sec.
LOIC TCP DoS Attack	19min./15sec.	19min./31sec.

Fig. 3 presents the response time of the distributed IDS system for various Nmap scans targeting the second and the third defense zone under different conditions of the infrastructure concerning the CPU load and the network throughput. The network throughput of the management and tunneling network were 900 Mbits/sec. and 1,02 Gbits/sec. respectively with deviation of 30 Mbits/sec. The results show that there is a pattern of metrics which could be used by security administrators in order to improve the security mechanisms. Such an improvement would take place by modifying the IDS sensors' ruleset and comparing the response time with the most often identified scans. Moreover, additional metrics as described on [31] should be taken into consideration so as to configure the IDS.

Finally, all attempts for DoS attacks created from systems of the third defense zone targeting the second defense zone systems, failed due to the security groups protection. Moreover, DoS attacks from the second defense zone targeting computing systems on the outside network, failed due to the first firewall. On the other hand, the success of an external DoS attack is highly defendant by the capacity of the targeted availability zone. However such an attack is confronted by the orchestration of availability zones.

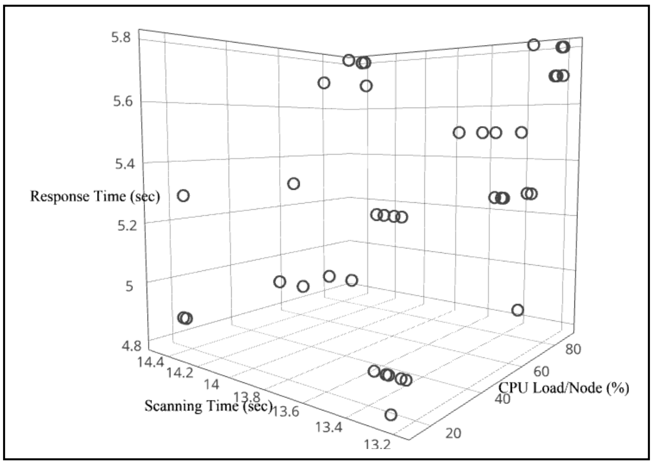

Fig. 3. 3D plot for the scanning and response time of Nmap scans under different CPU loads.

VII. Conclusions

Cloud computing is one of the most rapidly growing and adopting technology by organizations and companies which is threaten by a great number of attacks. The purpose of this paper is to define a security architecture based on defense in depth identifying the significance of protection of data alongside the productivity of the underline infrastructure. The main objective of the presented architecture was to ease the task of security for the newly migrated environments to cloud computing with minimum sacrifices in terms of scalability and availability. Apart from this, by adapting this security architecture into the cloud computing environment, the most dangerous threats are mitigated by a number of security systems that work with the same objective.

Acknowledgment

The publication of this paper has been partly supported by the University of Piraeus Research Center (UPRC) and the Technological Educational Institute of Western Macedonia.

References

[1] Gruber R., *Perimeter Security: Deter, Detect, Delay, Deny.* Master Halco Security Solutions Group, Available at: http://psi.praeger.com/pdfs/whitepapers/PerimeterSecurityandtheFourDs.pdf (Accessed at 21/12/15).

[2] Eman M., Hatem A. and Sherif E.-E., (2012). *Enhanced Data Security Model for Cloud Computing.* 8th international Conference on Informatics and Systems (INFOS), 14 -16 May, Cairo, pp. 12-17, ISBN: 978-1-4673-0828-1.

[3] Jianhua C., Yamin D., Tao Z. and Jie F., (2011), *Study on the security models and strategies of cloud computing*. Procedia Engineering, 23, pp. 586-593, ISSN: 1877-7058.

[4] Dan L., (2015). *Unobtrusive Intrusion Detection in Openstack.* Openstack Summit, Vancouver, BC May 2015

[5] K. M. Vieira, F. Schubert, G. A. Geronimo, R. de Souza Mendes and C. B. Westphall (2014). *Autonomic intrusion detection system in cloud computing with big data*. The 2014 International Conference on Security and Management, July 21-24, Las Vegas, USA, pp. 173-178.

[6] OpenStack Foundation, *OpenStack Kilo 2015.1.1*, Available at: https://www.openstack.org/

[7] Computer Security Division, Information Technology Laboratory, (2004). *Standards for Security Categorization of Federal Information and Information Systems.* National Institute of Standards and Technology (NIST), Federal Information Processing Standards Publication, pp. 5-8.

[8] Brough D., (2003), *Second Generation Honeynet Honeywall*. GIAC Security Essentials Certification, SANS Institute, pp. 2-14.

[9] Michael P. Brennan, (2002). *Using Snort For a Distributed Intrusion Detection System*. SANS Institute, pp. 2-12.

[10] Veerraju G., Srilakshmi I. and Satish M., (2012), *Data Security in Cloud Computing with Elliptic Curve Cryptography*. International Journal of Soft Computing and Engineering, 2(3), July 2012, pp. 138-141, ISSN: 2231-2307.

[11] Clad R., (2011). *Practical OSSEC*. GIAC Gold Certification, SANS Institute, pp. 2-30.

[12] Trustwave SpiderLabs. *ModSecurity*. Open Source Web Application Firewall, Available at: www.modesecurity.org

[13] Ramya. R, (2015). *Securing the system using honeypot in cloud computing environment*. International Journal of Multidisciplinary Research and Development, 2(4), April 2015, pp. 172-176, ISSN: 2349-5979.

[14] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, (1996). *Address Allocation for Private Internets*. Internet Engineering Task Force, Network Working Group, pp. 1-8.

[15] F Gont, (2012). *Deprecation of ICMP Source Quench Messages*. Internet Engineering Task Force, May 2012, pp. 2-5, ISSN: 2070-1721.

[16] F. Gont, (2010). *ICMP Attacks against TCP*. Internet Engineering Task Force, July 2010, pp. 1-31, ISSN: 2070-1721.

[17] *Spamhaus Project*, Available at: https://www.spamhaus.org/drop/

[18] Cisco Systems, (2005). *Cisco PIX Firewall and VPN Configuration Guide Version 6.3.*

[19] Open Web Application Security Project, *ModSecurity Core Rule Set Project*, Available at: https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project

[20] Ivan R., (2010). *Modesecurity Handbook The Complete Guide to Securing Your Web Applications*. Feisty Duck, pp. 143-157, ISBN: 1907117024.

[21] Rik B., (2010). Effectiveness of Defense Methods Against DDoS Attacks by Anonymous. University of Twente.

[22] M. Handley, E. Rescorla and Internet Architecture Board, (2006). *Internet Denial-of-Service Considerations*. Internet Engineering Task Force, Network Working Group, November 2006, pp. 1-30.

[23] W. Eddy, (2007). TCP SYN Flooding Attacks and Common Mitigations. Internet Engineering Task Force, Network Working Group, August 2007, pp. 1-13.

[24] The Snort Project, (2015). Snort Users Manual 2.9.8.0. November 18 2015, pp. 40-239.

[25] *DDOSim*, Available at: http://sourceforge.net/projects/ddosim/

[26] *R-U-Dead-Yet*, Available at: https://packetstormsecurity.com/files/97738/R-U-Dead-Yet-Denial-Of-Service-Tool-2.2.html

[27] *LOIC*, Available at: http://sourceforge.net/projects/loic/

[28] Fyodor, *NMAP,* Available at: https://nmap.org/

[29] Tenable, *Nessus,* Available at: http://www.tenable.com/products

[30] *Tcpdump*, Available at: http://www.tcpdump.org/

[31] Tim P., (2013). *How Can You Build and Leverage SNORT IDS Metrics to Reduce Risk?.* Infosec Reading Room, SANS Institute, 19 August 2013, pp. 11-14.

[32] Top Threats Working Group, (2013). *The Notorious Nine Cloud Computing Top threats in 2013.* Cloud Security Alliance, February 2013, pp. 8-21.