International Journal of Research in Computer Science
eISSN 2249-8265 Volume 4 Issue 1 (2014) pp. 27-34
www.ijorcs.org, A Unit of White Globe Publications
doi: 10.7815/ijorcs.41.2014.078

SCALABLE HONEYNET BASED ON ARTIFICIAL INTELLIGENCE UTILIZING CLOUD COMPUTING

Nogol Memari
Department of Computer and communication systems Engineering, Faculty of Engineering, UPM, MALAYSIA
Email: nogolmemari@gmail.com

Abstract: *The Honeynet is not a single system but a network sits behind a firewall where all inbound and outbound data is contained, captured and controlled. The Honeynet has two main components, data control and data capture. Data control is the way of filtering and allowing data flow and data capturing is collecting information for analyzing later on. The analysis is using the information which is collected and stored in Honeynet. The presence of high traffic makes it very difficult for human supervisors to monitor the status of the system 24/7, whereas by using an AI this task can be simplified. AI can monitor and analyze the hacker's activities, tools, IP addresses and adjust the Honeynet accordingly. Combination of Honeynet with AI based scaling and monitoring can indeed be very useful, both in terms of security and hardware resources.*

Keywords: *Artificial intelligence, Internet security, Cloud computing, scalability, Honeynet, Honeypot*

I. INTRODUCTION

The security over the internet is of particular interest, it is designed to give the users and organizations a degree of confidence in the system as the Internet represents a mostly insecure means of exchanging data and information between individual computers and mainframes leading to a high risk of intrusion or receiving fraud data. Most of the attackers and hackers exploit internet for financial gain, such as an industrial espionage to receive intellectual property such as patent information, sensitive employee information or sensitive customer information or they can use it as a means of sabotage for disgruntlement, such as an employee might be thinking about quitting or they might know that are about to be let off the job and might want to damaging another employee's or employers work [1, 2].

A computer represents a desirable target for the hackers, most non-military computer systems are not protected properly, and they are mostly missing decent firewall and internet security countermeasures. This coupled with recent advances in broadband internet connectivity protocols has led to a steady increase in number of attacks recorded, as the hackers can use automatic network scanners to scan a variety of networks simultaneously and thanks to the increase of connection speeds during the past decade, they easily access the files stored in the victims computer or upload special software to able them to utilize the resources of the victim's computer to their advantages. Also recently a spike in the number of malwares which uses the computers internet connection to send out junk e-mails or use their pc as a hub for peer to peer file sharing networks is clearly visible [3, 4].

The internet security can be divided into two main eras, the era of defensive security which began by the introduction of firewalls in early 90s which tried to block the unwanted internet connections, then during the mid-90s came the Intrusion Detection Systems (IDS). These defensive measures were mostly unsuccessful against committed hackers as they had the upper hand and all that security researchers could do was to guess the methods and resources used by the hackers and try to counter them. Then during the early 2000s came the honeypot, which tried to deceive and interact with the hacker and let him use his/her methods and resources trying to hack the honeypot, then this information would be used by the people responsible for the network security to adjust the security much more efficiently. The latest development in the network and internet security is Honeynet [5, 6].

II. HONEYPOTS AND HONEYNETS

A. Honeypot

Honeypots begin appearing during early 2000s, they were designed to present themselves an easy target and point of entry to a network. On the contrary they were mostly isolated and had a controlled input from the network, they disguised themselves as a legitimate part of the network and hoped to imitate the hacker to hack them, thus their methods and the malwares used could be recorded and analyzed so that appropriate countermeasures could be implanted with efficiency. These honeypots later evolved to what is currently known as Honeynet [7].

B. Honeynet

A Honeynet is a network of interconnected honeypots. A Honeynet is a type of decoy put in the servers, which mimics the real processes and services carried out by the real server. It is designed to be hacked and provides different levels of interactions with the hacker. A reverse firewall that captures all inbound and outbound data is used while deploying a Honeynet. The data gathered by this firewall is contained, captured, and controlled [6, 8].

C. Honeynet generations

There are three types of Honeynet, each providing different levels of interactions with the hackers, high and low interaction Honeynet are shown in Figure 1.

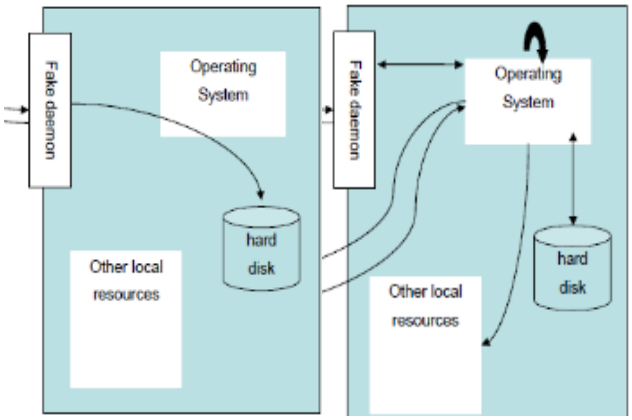

Figure 1: Low interaction (left) and high interaction Honeynet (right) [6].

The firewall controls how the Honeynet can initiate connection to internet. This factor prevents hackers from using the Honeynet to attack or compromise other production system on trusted networks. The Honeynet and the administrative network have no communication. There is the problem of hackers discovering that a firewall is filtering their traffic and the Honeynet is considered a success if the hackers never realize they were on Honeynet. One method for hiding firewall is using router. Firstly the router screens the Honeynet from the firewall. Once a Honeynet is compromised, the hacker will see a standard router instead of firewall. This is what they expect in most cases. Also the router acts as a second layer at access control. For example the router can be used for anti-spoofing control. Lastly the router can be used on additional layer of logging [9, 10].

It is also critical that anti-spoofing measures be properly implemented. Spoofing occurs from a different system or network. Spoofing the source IP address also makes it more difficult to track down and identify the attacker. Anti-spoofing ensures that only valid packets leave your Honeynet network.

Researches design computers or servers with Honeynet networks specifically to be attacked. They are configured to capture a variety of useful data about activities of the attacker inside the server and record all their requests and IP addresses. Researchers try to implement data capture and control in such a way that intruders remain unaware that their actions are being monitored. Honeynet are a useful tool for learning about computer intruder's tools, tactics, and motives. Implanting and maintaining a successful Honeynet requires attention to two critical elements, Data control and Data capture [11, 12].

Data control is filtering of what data flows where. The critical element is controlling what connections can be initiated outbound. To minimize risk, the system must be ensured that it cannot be used to attack production system on other networks. The key to data control is the use of an access control service, such as firewall. A firewall is used to separate the Honeynet from primary operational subsystems. [13].

Data capture is the collection of information, which is the end goal of Honeynet. The key for data capture is layers. A variety of techniques should be used to collect data.no layer should be a single point of failure. We use four layers namely access control layer, network layer, system layer and off-line layer. Although honey nets have many advantages such as keeping the hacker busy as he is wandering around and revealing his tactics, they can be taken over by the hacker and used as bots to attack other systems on the server. They also require a quite amount of system resources especially in the high interaction type [14].

III. CLOUD COMPUTING

Cloud computing is a relatively new concept in the IT industry. Due to recent advances in the fiber optic telecommunications and ever decreasing cost of powerful main frame computers, the idea of having all your data in one place and the ability of doing processes required on them as on the fly basis and accessing the processed information anytime and anywhere conveniently was very tempting for big organizations, this led to the birth of big main frame computers [15-18]. Some of these organizations such as Google or Amazon had huge data centers and began renting out the resources and processing power to any one requiring them [19]. This had a huge impact on the businesses as even small companies could own and operate an always accessible powerful main frame and could expand their computational resources with ease and a fraction of the cost of buying a new server in just minutes, and the term cloud computing was born and became popular. [20] In recent year the cost of having a cloud server is so decreased that even students can afford to rent one for their files and processing requirements [17, 20, 21].

The cloud computing undoubtedly provides organizations with the opportunity to save money and achieve efficiency, by enabling them to centralize applications, storage and platforms into unified platform which can be extended or reduced affectively with minimum cost. On the other hand without security embedded into underlying technology that supports cloud computing, businesses would be risking all their data [19, 22, 23].

Moving most or all of your data to an internet or network based cloud to save on costs introduces new risks to the organizations in pair with their existing security risks. In the attempt to provide cost savings, it is becoming increasingly common for virtual and physical security to be neglected. The cloud computing provides hackers with a golden opportunity, all the data they can dream of, gathered in one place, as many companies use a single cloud server to store and process all their data. This calls for many security measures to be taken, Honeynet being one of the most recent security measures [24-28].

Virtual Machines (VM) are a software platform developed due to recent advances in the processor and hardware field. It is designed to enable the user to run different operating systems simultaneously on the same hardware platform, utilizing every bit of resources possible to maximize efficiency [29]. Also they enable the quick switching between different operating systems efficiently without having to change the hardware or restart the system. [30] VMs are installed on a computer running a stable operating system (usually Linux or UNIX), this is called the host OS and multiple VMs can be installed on the same host OS depending on the hardware resources. One of the positive points of having VM installed is that it can be configured to represent a complete standalone hardware platform with its own firewall, ram, hard disk, processor, network connection and IP address. This is particularly useful in cloud based services as we can configure the preferences of each VM on demand and rent it to the requesting individual or organization [31-33].

IV. DESIGN OF A HONEYNET

No single product, method or solution for a Honeynet exist, it all depends on requirements and environment. However the functionality of data control and data capture must be met. Regardless of what architecture developed and implemented, you must be able to both control and capture data. Honeynet is not a deploy and forget solution. They require constant care and feeding. One example is using a single firewall to segment the Honeynet to three distinct networks: the internet, the administrative trusted network and the Honeynet [34, 35].

DNS (domain name system) resolution and NTP (network time protocol) services are also needed. DNS is a required functionality, as the hackers often rely on DNS resolution for tool download or activation. NTP ensures that all system time clocks are synched. This is helpful for data from various systems is on the same time [36].

A. Access control layer

The first layer is access control such as firewalls or routers. Any packet entering or leaving the Honeynet must pass through these devices as they are an excellent resource of information .for most organizations, telnet, RPC (remote processing call) and ICMP (internet control message protocol) utilization is normal. It can be difficult to distinguish between an innocent RPC request and a malicious. The Honeynet solves that problem by flagging any data that either enters or leaves the network. There are alerts for outbound and inbound connections [37].

B. Network layer

This layer collects two types of information: suspicious signature alerts and packet payload. The first is the alerting to suspicious activity. These alerts inform the administrator what is happening in real-time. The second is the capturing of every packet and the packet's payload that enters and leaves the network. This information is stored in a log file and the data can be retrieved later date for detailed analysis. The third layer of data is any ASCII payload, such as keystrokes or IRC sessions that are stored in separate ASCII flat file [38].

System layer used for sorting information remotely on a protected server .Off-line layer creates images of compromised system, so that you can conduct an offline analysis of the system and determine what the hacker did.it is possible to reconstruct the hacker's activity even without the keystrokes [39].

V. CHALLENGES IN HONEYNET

Although Honeynet are among the most powerful tools to analyze the intruder's activities on the network, they do come with a price. To obtain the information on the hacker's activity, a certain level of access to the systems and applications must be provided so the intruder could be tricked in believing that he is attacking a vital system. Once the true identity of a Honeynet has been exposed to the hacker, its value is dramatically reduced. Attackers can ignore or bypass the Honeynet, eliminating its capability for capturing information. Also the attacker can implant false data in attacking the Honeynet thus the data analysis would be useless or misleading. The

Honeynet is not a single system but a network which sits behind a firewall where all inbound and outbound data is contained, captured and controlled. The presence of high traffic makes it very difficult to monitor the status of the system 24/7. The raw processing power and hardware requirements are high, but most of the time these are not required and the system can run on much less. To decrease the hardware requirements of Honeynet, we can employ Honeynet on virtual machines. [40].

The virtual machine environment offers several benefits and is an ideal base for implementing a Honeynet. First, it is easy to manage, since most virtual machine managers (VMM) allow individual VMs to be enabled, hibernated or the whole system to be saved on demand. Advantage of having such a control over the installed VMs is the driving force behind the use of VMs in Honeynet in operational deployments. As well, VMs also offer an ideal platform for monitoring and storing the activities within a compromised Honeynet, including interactive input, memory and disk allocation, patterns of system calls and the content of endpoint network. Finally, VMs allow multiple Honeynet to be implemented by a single hardware base such as a single mainframe thus reducing deployment costs.

On the hardware pint of view, as each Honeynet has a single IP address and since this IP address is accessed rarely, much of the processing power dedicated to the Honeynet remains unused. Even when serving a request, most of a honeypot's memory is idle as well. Finally, different honeypot servers in a Honeynet replicate the same environment and thus duplicate the effort in maintaining common state and executing common code paths. In fact, a conventional Honeynet network will use far less than one percent of processor or memory resources for their intended purpose. Given this waste of resources which can be used more efficiently on other tasks, an artificial intelligence based approach to scale the number of VMs in a Honeynet is proposed [41, 42].

The objective of the study is to design and implement an Artificial Intelligence (AI) based scaling method to minimize the utilization of the resources on the server dedicated to the Honeynet while maximizing the efficiency of the hardware based processes. This approach will free the hardware resources which could be used for processing of the requests while not compromising the security of the server as Honeynet will become active on presence of high and suspicious network traffic and will deactivate when there is no suspicious network traffic. By use of virtual machines and virtual machine managers and implanting a router based gateway, the system can be monitored all the time and the necessary adjustments can be done autonomously by an AI [34, 36, 43].

VI. METHODOLOGY AND RESULTS

First thing to do is to implement the virtual machine environment, to do so we install a virtual machine manager and multiple virtual machines in our server. After setting up our virtual machines, we then turn our attention to the gateway. Gateway enables us to remove the idleness in our IP zone utilization, also the use of a gateway enables the us to use isolated operating systems (as VMs) on a single server, this eliminates the risk of the hacker intruding all the VMs and Honeynet installed on those VMs with a single attack, as the attacker needs to repeat the same attack on each individual VM. The essential part of the scaling comes from the gateway, as all the data entering and exiting the server must go through the gateway, we can use the information obtained from the gateway to successfully scale our Honeynet [32, 44].

One of the positive point of VMs is the ability to start and terminate them on demand, this is achieved by implanting a virtual machine manager, now the data obtained from the router can be used to determine the number of Honeynet required to ensure the security of our system and keep the hackers as busy as possible, the gateway will assign IP addresses as required by the VMM and remove the IP address when the VM associated with that address is no longer active. This approach enables us to multiplex our hardware capabilities and use ore resources to the best extend [30, 45, 46].

Cloud security is important because many services such as mail servers, accounting and information servers are stored in the cloud. So that each day many hackers from internet or internal staff try to get access by malware such as bots and spyware to misuse information .That is why many security measures are taken; Honeynet works by showing a decoy to the hacker and presenting themselves as the services subsystem. As these honey net use system resources, it is desirable to scale the virtual machines to allocate the lowest level of resources required at the time .On the contrary the attacks vary from time to time that is why we use an attack analyzer to monitor the level of activity on the honey net and the information from the analyzer is fed to the expert system which in turn adjusts the level of resources dedicated to the virtual machines. Figure 2 represents the flowchart of the proposed method.

There are three primary means of data analysis; the first method of information analysis is firewall alerts, which give us real-time information on the hackers activity .in addition, this activity is achieved for future use. The second layer and one of the most important is packet capture. Every packet and information is stored in both binary and ASCII formats. Suspicious activity can also be detected when captured on the wire. The

last layer for data analysis, the system logs, tells us activity the system [47].

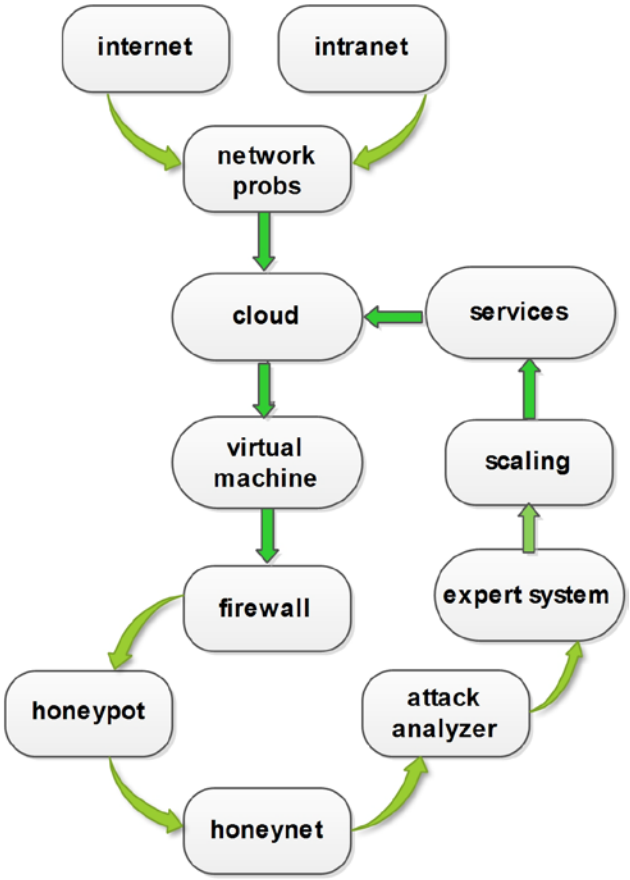

Figure 2: Diagram of proposed framework of scalable Honeynet.

There are three primary means of data analysis; the first method of information analysis is firewall alerts, which give us real-time information on the hackers activity .in addition, this activity is achieved for future use. The second layer and one of the most important is packet capture. Every packet and information is stored in both binary and ASCII formats. Suspicious activity can also be detected when captured on the wire. The last layer for data analysis, the system logs, tells us activity the system [47].

There are two techniques for more advanced data analysis. Passive finger printing demonstrates how information can be passively gathered from packets sent by the remote system. This allows discovery of important information, such as identification of the remote operating system or the application being used [47, 48].

Although this information may not seem important, small bits of information pieced together can prove critical in assembling the big picture. Forensics is a second, and for more involved, technique of data analysis. All Honeynet systems are designed with forensic analysis in mind.

Forensic analysis is the process of recovering, capturing and analyzing information from compromised coroner's toolkit is the preferred weapon of choice for the Honeynet project when analyzing Unix-based systems. The TCT tools are the primary tolls used for data recovery and analysis [49, 50]. En example of Honeynet response to incoming network probes can be seen in the following paragraph.

A sample of windows7 was emulated inside Honeynet by low interaction honeypot named Honeyd, a script for port 80 was written to present a real looking webserver, then an IP address and Mac address was assigned to this emulated windows and some port were considered to be open .An automatic network probing tool called Nessus was used to probe the network to ensure the designed framework was working properly.

The virtual machine is connected to internet via port 6600 on the server and traffic is routed to port 80 on the virtual machine.

Emulation of a Windows 7 in Honeynet can be accomplished by following configurations:

Create Windows

set Windows personality "Windows 7"

set Windows default tcp action reset

set Windows default udp action reset

set Windows default icmp action open

add Windows tcp port 25 open

add Windows tcp port 80

"/usr/share/honeyd/scripts/web.sh"

add Windows tcp port 110 open

add Windows udp port 123 open

add Windows udp port 135 reset

add Windows tcp port 137 reset

add Windows udp port 137 reset

add Windows udp port 138 open

add Windows udp port 139 open

add Windows tcp port 143 open

add Windows tcp port 143 open

add Windows udp port 445 open

add Windows udp port 500 open

add Windows udp port 1900 open

add Windows udp port 4500 open

add Windows udp port 31337 open

set Windows uptime 1336262

set Windows ethernet "00:00:2a:a8:8a:3a"

bind 192.168.1.5 Windows

Fingerprint for windows7 that was sent in response to network probe for emulated windows7 can be seen below:

FingerPrint Windows 7

TSeq(Class=TR%IPID=I%TS=100HZ)

T1(DF=Y%W=2000%ACK=S++%Flags=AS%Ops=M NWNNT)

T2(Resp=Y%DF=Y%W=0%ACK=S%Flags=AR%Ops=)

T3(Resp=Y%DF=Y%W=0%ACK=O%Flags=AR%Ops =)

T4(DF=Y%W=0%ACK=O%Flags=R%Ops=)

T5(DF=Y%W=0%ACK=S++%Flags=AR%Ops=)

T6(DF=Y%W=0%ACK=O%Flags=R%Ops=)

T7(DF=Y%W=0%ACK=S++%Flags=AR%Ops=)

PU(DF=N%TOS=0%IPLEN=164%RIPTL=148%RID= E%RIPCK=E%UCK=E%ULEN=134%DAT=E

The report from network probe can be seen in the following paragraph, it can be seen that the framework performed as designed and the probe identified the virtual machine operating system as windows7.

Nmap Scan Report - Scanned at Tue Dec 03 19:30:43 2013 Scan Summary

Nmap 6.40 was initiated at Tue Dec 03 19:30:43 2013 with these arguments:

nmap -p T:6600 -O 172.16.71.159

172.16.71.159(online)

Ports 6600 tcp open

Remote Operating System Detection

- Used port: 6600/tcp (open)
- Used port: 44221/udp (closed)
- OS match: Microsoft Windows 7 SP1 (88%)
- OS match: Microsoft Windows Vista SP0 - SP1 (88%)
- OS match: Microsoft Windows 7 (87%)

On the following line it can be seen that Honeynet recorded the probe with source IP of probing computer and port number used for the probe.

Honeynet Log:

2013-12-03-19:30:46.4248 tcp (6) S 172.16.71.154 34789 192.168.1.5 80

VII. CONCLUSION & FUTURE WORK

The Honeynet is not a single system but a network that sits behind a firewall where all inbound and outbound data is contained, captured and controlled. The Honeynet is implementation has two issues: data control and data capture. Data control is the way of filtering and allowing data flow and data capturing is collecting information for analyzing later on. Data capture is implemented by four layers: access control, network layer, system layer and off-line layer. The analysis is using the information which is collected and stored in Honeynet. Although Honeynet have many advantages such as keeping the hacker busy as he is wandering around and revealing his tactics, they can be taken over by the hacker and used as bots to attack other systems on the cloud server. They also require a quite amount of system resources especially in the high interaction type.

The ability of introducing new Honeynet to the system as required by the circumstances at the time enables us to maximize the potential of gathering data and keep the required hardware recourses as minimal as possible. Future works include hardening and making the whole Honeynet system more robust while decreasing the chance of detection by the intruder. Also more robust implementation of expert system for controlling the scalability aspect of the Honeynet is required as many intruders and hackers are using automated network probes which scan a wide variety of networks and report the suspicious and easily accessible ones to the intruder, the implemented expert system needs to identify and respond to these probes accordingly without the need to use high amounts of system resources.

VIII. REFERENCES

[1] Michael E. Whitman, & Herbert J. Mattord, “Principles of information security. Cengage Learning”, 2010. ISBN: 1111138214

[2] Charles Haley, Robin Laney, Jonathan Moffett, Bashar Nuseibeh, "Security Requirements Engineering: A Framework for Representation and Analysis," IEEE Transactions on Software Engineering, vol. 34, no. 1, pp. 133-153, January, 2008. doi: 10.1109/TSE.2007.70754

[3] M. Howard, S. Lipner, “The security development lifecycle”, Microsoft Press, vol. 11, 2009. ISBN: 9780735622142

[4] B. R. Kandukuri, V.R. Paturi, A. Rakshit, “Cloud security issues”, IEEE International Conference on Services Computing, pp. 517-520, 2009. doi: 10.1109/SCC.2009.84

[5] Daniel E. Geer, "The evolution of security", queue. 5, no.3, pp.30-35, 2007. doi: 10.1145/1242489.1242500

[6] Niels Provos, Thorsten Holz, "Virtual honeypots: from botnet tracking to intrusion detection", Pearson Education, 2007.

[7] J.S Bhatia, R. Sehgal, B. Bhushan, H. Kaur, "A case study on host based data analysis & cyber criminal profiling in Honeynets", First International Communication Systems and Networks and Workshops, pp.1-2, 2009. doi: 10.1109/COMSNETS.2009.4808902

[8] Bahman Nikkhahan, Akbar Jangi Aghdam, Sahar Sohrabi, "E-government security: A honeynet approach", International Journal of Advanced Science and Technology, vol.5, 2009.

[9] Davide Cavalca, Emanuele Goldoni, "An open architecture for distributed malware collection and analysis", Open Source Software for Digital Forensics, Springer US, pp. 101-116, 2010. doi: 10.1007/978-1-4419-5803-7_7

[10] C.C. Zou, R. Cunningham, "Honeypot-aware advanced botnet construction and maintenance", International Conference on Dependable Systems and Networks, pp.199-208, 2006. doi: 10.1109/DSN.2006.38

[11] D. Watson, J. Riden, "The honeynet project: Data collection tools, infrastructure, archives and analysis", WOMBAT Workshop on Information Security Threats Data Collection and Sharing, WISTDCS'08, pp.24-30, 2008, doi: 10.1109/WISTDCS.2008.11

[12] Y. Zhou, J. Zhuge, N. XU, X. JIAO, W. SUN, Y. JI, Y. DU, "Matrix, a distributed honeynet and its applications", 20th Annual FIRST Conference (FIRST'08), Canada, 2008.

[13] Young Hoon Moon, Huy Kang Kim, "Proactive Detection of Botnets with Intended Forceful Infections from Multiple Malware Collecting Channels", Future Information Technology, Springer Berlin, pp.29-36, 2011. doi: 10.1007/978-3-642-22333-4_4

[14] C. Leita, V. H. Pham, O. Thonnard, E.S. Ramirez, F. Pouget, E. Kirda, M. Dacier, "The leurre. com project: collecting internet threats information using a worldwide distributed honeynet", WOMBAT Workshop on Information Security Threats Data Collection and Sharing, WISTDCS'08, pp.40-57, 2008. doi: 10.1109/WISTDCS.2008.8

[15] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "A view of cloud computing", Communications of the ACM, 53(4), pp.50-58, 2010. doi: 10.1145/1721654.1721672

[16] Qi Zhang, Lu Cheng, Raouf Boutaba, "Cloud computing: state-of-the-art and research challenges", Journal of Internet Services and Applications, vol.1, issue1, pp.7-18, 2010. doi: 10.1007/s13174-010-0007-6

[17] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation computer systems, vol.25, issue.6, pp.599-616. doi: 10.1016/j.future.2008.12.001

[18] Peter Mell, Timothy Grance, "The NIST definition of cloud computing", NIST special publication, 800(145), 2011.

[19] Katarina Stanoevska-Slabeva, Thomas Wozniak , "Cloud basics–an introduction to cloud computing", Grid and Cloud Computing, Springer Berlin Heidelberg, pp.47-61, 2010. doi: 10.1007/978-3-642-05193-7_4

[20] Hui Jie Ding, "Traffic Flow Data Collection and Signal Control System Based on Internet of Things and Cloud Computing", Advanced Materials Research, vol.846, pp.1608-1611, 2013. doi: 10.4028/www.scientific.net/AMR.846-847.1608

[21] Nuno Santos, Krishna P. Gummadi, Rodrigo Rodrigues, "Towards trusted cloud computing", Conference on Hot topics in cloud computing, pp.3, 2009. doi: .

[22] Peter Mell, Tim Grance, "Effectively and securely using the cloud computing paradigm", NIST, Information Technology Lab, 2009.

[23] L. Wang, J. Tao, M. Kunze, A.C. Castellanos, D. Kramer, W. Karl, "Scientific cloud computing: Early definition and experience", 10th IEEE International Conference on High Performance Computing and Communications, HPCC'08, pp. -830, 2008. doi: 10.1109/HPCC.2008.38

[24] Bernd Grobauer, Thomas Schreck, "Towards incident handling in the cloud: challenges and approaches", ACM workshop on Cloud computing security workshop pp.77-86, 2010. doi: 1 0.1145/1866835.1866850

[25] Chang-Lung Tsai, Uei-Chin Lin, A.Y. Chang, Chun-Jung Chen, "Information security issue of enterprises adopting the application of cloud computing", Sixth International Conference Networked Computing and Advanced Information Management, pp.645-649. 2010.

[26] Sean Carlin, Kevin Curran, "Cloud computing security", International Journal of Ambient Computing and Intelligence (IJACI), vol.3. no.1, pp.14-19, 2011.

[27] Yanpei Chen, Vern Paxson, Randy H. Katz, "What's new about cloud computing security", University of California, Berkeley Report No. UCB/EECS-2010-5, 2010.

[28] Borko Furht, Armando J. Escalante, "Handbook of cloud computing", Springer Publishing Company, Incorporated, 2010.

[29] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, Andrew Warfield, "Remus: High availability via asynchronous virtual machine replication", Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, pp.161-174, 2008.

[30] K. Nance, B. Hay, M. Bishop, "Investigating the implications of virtual machine introspection for digital forensics. International Conference on Availability, Reliability and Security, ARES'09, pp.1024-1029, 2009. doi: 10.1109/ARES.2009.173

[31] Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, Anthony Liguori, "KVM: the Linux virtual machine monitor", Proceedings of the Linux Symposium, vol.1, pp.225-230, 2007.

[32] F. Lombardi, R. Di Pietro, “Secure virtualization for cloud computing”, Journal of Network and Computer Applications, vol.34, no.4, pp.1113-1122, 2011.

[33] eanna Matthews, Tal Garfinkel, Christofer Hoff, Jeff Wheeler “Virtual machine contracts for datacenter and cloud computing environments”, Proceedings of the 1st workshop on Automated control for datacenters and clouds, pp. 25-30, 2009. doi: 10.1145/1555271.1555278

[34] J. Quan, K. Nance, Brian Hay, “A Mutualistic Security Service Model: Supporting Large-Scale Virtualized Environments” IT Professional, vol.13, no.3, pp.18-23, 2011. doi: 10.1109/MITP.2011.36

[35] Jun Wang, Jing Zeng, “Construction of large-scale honeynet Based on Honeyd”, Procedia Engineering, vol.15, pp.3260-3264, 2011. doi: 10.1016/j.proeng.2011.08.612

[36] M.H. Sqalli, S.N. Firdous, Z. Baig, F. Azzedin, “An Entropy and Volume-Based Approach for Identifying Malicious Activities in Honeynet Traffic. International Conference on Cyberworlds (CW), pp.23-30, 20011 doi: 10.1109/CW.2011.35

[37] Olivier Thonnard, Marc Dacier , “A framework for attack patterns' discovery in honeynet data”, Digital investigation, vol.5, pp.S128-S139, 2008. doi: 10.1016/j.diin.2008.05.012

[38] Ping Wang, Lei Wu, Ryan Cunningham, Cliff C. Zou, “Honeypot detection in advanced botnet attacks”, International Journal of Information and Computer Security, vol.4, no.1, pp.30-51. doi: 10.1504/IJICS.2010.031858

[39] J.S. Bhatia, R. Sehgal, B. Bhushan, H. Kaur, “M"ulti Layer Cyber Attack Detection through Honeynet”, New Technologies, Mobility and Security, NTMS'08, pp.1-5, 2008. doi: 10.1109/NTMS.2008.ECP.65

[40] Yao Zhao, Yinglian Xie, Fang Yu, Qifa Ke, Yuan Yu, Yan Chen, Eliot Gillum, “BotGraph: Large Scale Spamming Botnet Detection”, Proceedings of the 6th USENIX symposium on Networked systems design and implementation (NSDI'09), vol.9, pp.321-334, 2009.

[41] Wira Zanoramy Ansiry Zakaria, S.R. Ahmad, Norazah Abd Aziz, “Deploying virtual honeypots on virtual machine monitor”, International Symposium on Information Technology, ITSim 2008, vol.4, pp.1-5. doi: 10.1109/ITSIM.2008.4631930

[42] Ting Zhang, Lin Hong Guo, “Research and Implementation of Experimental Platform for Network Attack and Defence Based on Honeynet”, Advanced Materials Research, vol.403, pp.2221-2224, 2012. doi: 10.4028/www.scientific.net/AMR.403-408.2221

[43] Jianwei Zhuge, Thorsten Holz, Xinhui Han, Chengyu Song, Wei Zou, “Collecting autonomous spreading malware using high-interaction honeypots”, Information and Communications Security, Springer Berlin Heidelberg , pp. 438-451, 2007. doi: 10.1007/978-3-540-77048-0_34

[44] Michael Ligh, Steven Adair, Blake Hartstein, Matthew Richard, “Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code”, Wiley Publishing, 2010.

[45] Michael Vrable, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage, “Scalability, fidelity, and containment in the potemkin virtual honeyfarm”, ACM SIGOPS Operating Systems Review, vol.39, no. 5, pp.148-162, 2005. doi: 10.1145/1095809.1095825

[46] George W. Dunlap, Samuel T. King, Sukru Cinar, Murtaza A. Basrai, and Peter M. Chen, “ReVirt: Enabling intrusion analysis through virtual-machine logging and replay”, ACM SIGOPS Operating Systems Review, vol.36(SI), pp.211-224.

[47] Zhichun Li, Anup Goyal, Yan Chen, “Honeynet-based botnet scan traffic analysis”, Botnet Detection, vol 36, pp.25-44, Springer US, 3008. doi: 10.1007/978-0-387-68768-1_2

[48] Krasser, Sven, G. Conti, J. Grizzard, J. Gribschaw, H. Owen, “Real-time and forensic network data analysis using animated and coordinated visualization”, Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, IAW'05, pp. 42-49, 2005. doi: 10.1109/IAW.2005.1495932

[49] F. Daryabar, A. Dehghantanha, F. Norouzi, F. Mahmoodi, “Analysis of virtual honeynet and VLAN-based virtual networks” International Symposium on Humanities, Science & Engineering Research (SHUSER), pp. 73-77, 2011. doi: 10.1109/SHUSER.2011.6008503

[50] Samuel Oswald Hunter, “Virtual Honeypots: Management, attack analysis and democracy”, March 2010.

How to cite

Nogol Memari, “Scalable Honeynet Based on Artificial Intelligence Utilizing Cloud Computing”. *International Journal of Research in Computer Science*, 4 (1): pp. 27-34, January 2014. doi: 10.7815/ijorcs.41.2014.078