Journal of Engineering and Applied Sciences 12 (Special Issue 2): 6208-6214, 2017
ISSN: 1816-949X

An Implementation of Honeypots in a Cloud Environment for Analyzing Attacks on Websites

K.S. Vishal, Sarthak Chauhan and K. Krishna Prakasha
Department of Information and Communication Technology,
Manipal Institute of Technology, Manipal University, Manipal, Karnataka, India

Abstract: We deploy honeypots over the cloud to gather information and analyse the attacks that try to illegitimately access websites. The methods and the malware used to attack websites has been continuously evolving, never has the need for good network security been more apparent than it is now. With multiple kinds of 0 day exploits and more attackers than ever before deploying honeypots will be an essential part of any network security setup due to their potential for catching attackers in the "act" and to find out who the attacker is. With honeypots we have a way to observe the malware that attackers use and isolate it before it can harm the system. In this study, we setup a wordpress blog website and gather information then analyze this data to gain insights about the current threats in the world of internet security.

Key words: Cloud, honeypot, network security, find out, system, security

INTRODUCTION

A honeypot is a tool of deception for attracting attackers to make efforts to gain access to the information systems of an organization. A honeypot serves as an advanced security tool for use in minimizing the risks of attacks on information technology systems and networks. Honeypots are useful for providing valuable insights into potential system security loopholes. Honeypots are useful for finding out the vulnerabilities a system has Olangunju *et al*. (2016).

The unique contributions of this research include: a demonstration how open source technologies are used to dynamically add or modify hacking incidents in a low-interaction honeynet system a presentation of strategies for making honeypots more attractive for hackers to spend more time to provide hacking evidences.

Problem definition: The persistent increase of cyber attacks is sending warning signals not only to security professionals but also to business managers who witness financial losses soaring due to these attacks resulting in a negative impact on their business-financial prosperity. There are numerous studies documenting the rise of security incidents and at the same time alerting for a significant additional number of successful breaches that go undetected.

According to the "the global state of information security survey" by PWHC (2014), the number of detected security incidents in 2014 was 48% up from 2013. There are several traditional approaches to deter cyber attacks, like setting perimeter security and defense in depth. However, the sophistication of the cyber attacks call for thinking outside the box and deploy active defense mechanisms that are within the legal rights of an organization to protect its assets. A proactive strategy allows for analyzing attacks and preparing countermeasure mechanisms to protect against them. A honeypot could be seen as a trap that lures in attackers in order to study their attack patterns. It must be configured and set up in a realistic manner so that attackers will direct their time, attention and energy toward something that is useless from an attack perspective. This form of deception which aims in observing the adversary in action, still remains underrepresented in the pool of defensive and protection mechanisms (Amoroso, 2010).

According to "proactive detection of security incidents: honeypots" other barriers to the wider deployment of honeypots include difficulty with usage, poor documentation and lack of software stability (Polska, 2012). However, there is a growing support towards honeypots from organizations like ENISA, advising that Computer Emergency Response Teams (CERTs) of national governments could benefit greatly by the wider adoption of honeypots (Dittrich, 2004). Threat monitoring is the process of scanning the network and endpoints for security threats. Threat monitoring is classified as host based and network

Corresponding Author: K.S. Vishal, Department of Information and Communication Technology,
Manipal Institute of Technology, Manipal University, Manipal, Karnataka, India

based. Most anti-virus software and intrusion detection systems are classified as host based (Chawda *et al*., 2014). These programs run in the background and continuously monitor the system for threats, if a malware or a Trojan is found it is immediately quarantined and seeks permission from user to delete it. They accomplish this by analyzing the behavior of malware and other malicious programs and sometimes known attacks have a 'signature' which makes them easier to identify. Host based honeypots work on the principle of making the honeypot as enticing as possible for the attacker so that he definitely attacks it (Spitzner, 2002). Network based honeypots offer the advantage of r eal OS services and applications which lend themselves to a more authentic experience for the attacker. It allows for the capture of extensive amount of information on the attacker's behavior on the compromised system the trade off being these systems are hard to deploy and maintain while the risk of collateral damage is also high as the compromised system could be used to attack other systems on the internet (Amoroso, 2010). Honeypots are meant to provide data on why the attacker is attacking, what his motives might be and most importantly what data he wants to get. Another approach is passive monitoring where the honeypot monitors the network, these minimally effect the working of the system and are light weight. Passive monitoring also falls into 3 categories, data from security or policy enforcement devices, data from traffic characterization mechanisms and direct sensing or sniffing (Chawda *et al*., 2014).

Objective: The objectives of this research are:

- To use free and open-source technologies and methods to reduce the amount of manual intervention required to add or modify a honeypot system
- To detect attack patterns on network system services

Scope: The research is supposed to benefit:

- Cyber security professionals
- Network security professionals
- Researchers in academia

Literature review: There are two types of honeypots. Low-level and high-level interaction honeypots, low-level honeypots are used to emulate services while high-level honeypots are used to emulate the entire operating system. Honeypots can be used for a multitude of tasks, to detect spammers, detect USB malware and even for database protection against sql injections and the like (Amoroso, 2010). There are two major requirements of any effective honeynet architecture, data control and data capture. Data control is needed to find out how the attack is taking place what data the attacker is using and his method of attack by targeting the flow of information. Data control is used to keep the attacker in the honeypot and not let him venture outside to attack legitimate targets (PWHC, 2014) data capture is used to identify critical information about the attacker and his attack vectors. Large organizations can implement honeypot technology to defend against Distributed Denial of Service (DDoS) attacks (Spitzner, 2002). A system that can defend the operational network of an organization against known DDoS and new future types of attacks can be setup. The system includes a Demilitarized Zone network (DMZ) that implemented services such as web, mail, ftp and DNS for access by external networks. A firewall is used to protect the local internal network of the organization in another zone. A honeypot is effectively used to mimic the internal network systems and attract DDoS attackers. If the attackers compromised packets to the web server of the corporation are detected; the packets go to the honeypot for processing. The attacker receives a reply that can be indistinguishable from the actual response from a web server.

The system is capable of trapping the attacker and recording the compromised components of the network to provide evidence for use in a legal action. The current research gained ideas from this study as we designed and implemented algorithms for detecting attacks, actively directing attack packets to the honeypots and making the honeypots to simulate the network infrastructure of an organization.

MATERIALS AND METHODS

The system consists of a Firewall, DataHero, LAMP Stack, the modern honey network project, 4 honeypots namely Elastichoney, Wordpot, Snort and pOf. Modern Honey Network (MHN) (Trost, 2017); MHN is a honeypot management system which enables the creation of a fully functional active-defense network. MHN and four honeypots run on virtual machines. Elastichoney is a simple elastic search honeypot designed to catch attackers exploiting RCE vulnerabilities in elastic search. Dionaea aims to trap malware exploiting vulnerabilities exposed by services offered over a network and ultimately obtain a copy of the malware. A combination of Snort and

PoF is used for data capture mode and Network Intrusion Detection System (NIDS) mode which performs detection and analysis on network traffic. DataHero is self-service cloud BI that allows users to quickly connect to cloud services for automatically updated insights. The entire network is hosted on a cloud instance from AWS. The LAMP stack is implemented. Ubuntu 14.04 is used as the OS, Apache as the web server, PHP and MySQL as the database. Apache web server is installed to respond to incoming requests. The webpage acts as a front-end for capturing attack information such as login ID and Passwords.

After the LAMP stack is installed, the site is secured using an SSL certificate. We generate keys on the web server and using Comodo SSL get a free SSL certificate to use with our domain, www.cyber-space.online.

After the SSL is installed the wordpress site is setup. To connect the MySQL database to the front-end. Start the MySQL server and create a user. Create a database and grant the user full rights to the database. This database holds information such as posts made on the website and users authorized to post on the website as well as comments posted. Changes are made in the wordpress configuration file to reflect the MySQL wordpress user changes. After this the wordpress site is ready to use.

Honeypot Software has been in development for quite some time now and this can be regarded as a maturing phase for the various kinds honeypot software as they still don't see wide deployment. A big reason for this is the fact that they are regarded as complicated to deploy and manage at high scales, especially in enterprise where the cost of deployment and maintenance will be quite high and with little to no visible advantages other than security.

The modern honey network makes deploying and management of honeypots easier. We can deploy and run honeypots with a simple click and see in real-time the deployment of our honeypots and the attackers IP. It is open-source and uses other open-source honeypots.

Installing MHN: To install MHN the following commands need to be entered in the host server's terminal:

- $ cd/opt/
- $ sudo git clone https://github.com/threatstream/mhn.git
- $ cd mhn/
- $ sudo./install.sh

Configuring MHN:

```
MHN Configuration
Do you wish to run in Debug mode?: y/n n
Superuser email: ******@gmail.com
Superuser password: ******
Server base url ["http://1.2.3.4"]: http://ec2-34-223-210-40.us-west-2.compute.amazonaws.com
Honeymap url ["http://1.2.3.4:3000"]: http://ec2-34-223-210-40.us-west-2.compute.amazonaws.com:3000
Mail server address ["localhost"]: localhost
Mail server port [25]: 25
Use TLS for email?: y/n n
Use SSL for email?: y/n n
Mail server username [""]:
Mail server password [""]:
Mail default sender [""]:
Path for log file ["mhn.log"]:
```

Setting up MHN: After this, the installation will start to download and load snort rules from emerging threats. When the rules have finished importing the installation is complete and we can visit the MHN console by going to the URL we gave during configuration.

Configuring MHN to use HTTPS: NGINX is a free, open-source, high-performance HTTP server and reverse proxy as well as an IMAP/POP3 proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration and low resource consumption.

Unlike traditional servers, NGINX doesn't rely on threads to handle requests. Instead it uses a much more scalable event-driven (asynchronous) architecture. This architecture uses small but more importantly, predictable amounts of memory under load. Even if you don't expect to handle thousands of simultaneous requests, you can still benefit from NGINX's high-performance and small memory footprint. NGINX scales in all directions: from the smallest VPS all the way up to large clusters of servers. This MHN server image has some pre deployed NGINX configuration files. To enable these now that MHN is installed we run these commands:

- cd /etc/nginx/sitesenabled
- ln s/etc/nginx/sitesavailable/mhnhttps
- ln s/etc/nginx/sitesavailable/honeymaphttps
- rm/etc/nginx/sitesenabled/default
- nginx restart

Honeypot deployment: After we setup the MHN server we deploy our honeypots on the wordpress blog the honeypots used are: Elasticsearch+Snort+pOf+Dionaea From the MHN server login page, we login using the credentials given during the MHN setup (Fig. 1 and 2). Select deploy from the menu and run the honeypot scripts on the wordpress web server to finish setting up the honeynet.

Now to integrate with DataHero to analyze the attacks on the website. Data needs to be exported from Snort. Snort is an open source intrusion prevention system capable of real-time traffic analysis and packet logging in a PCAP file format.

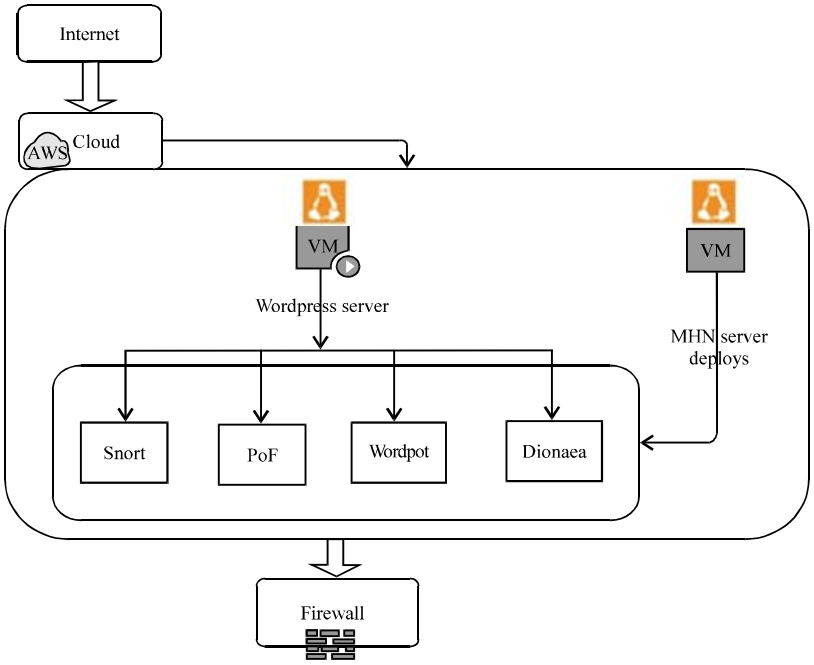

Fig. 1: Block diagram of proposed network architecture

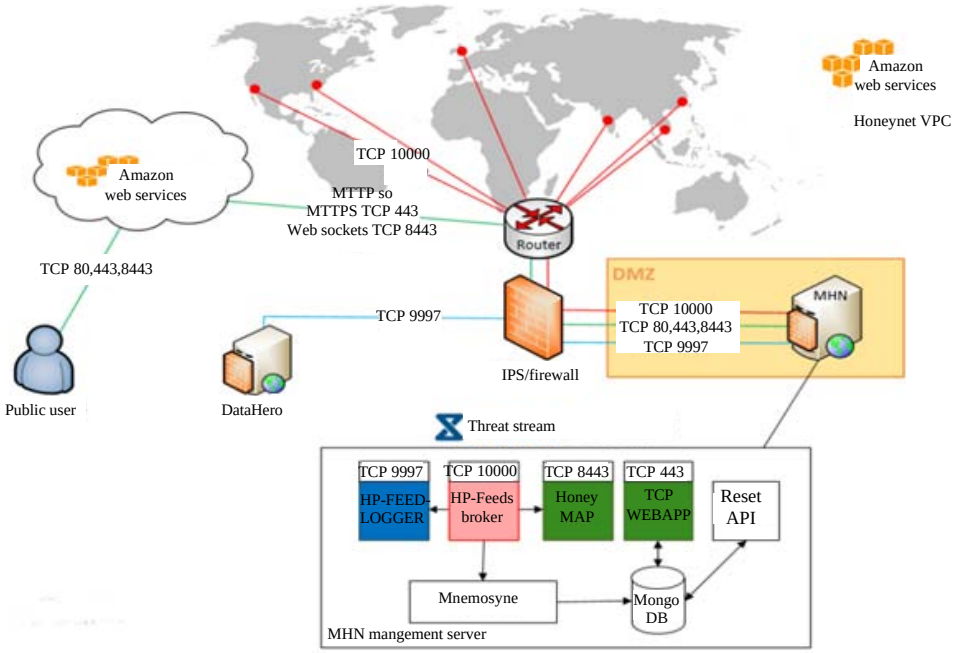

Fig. 2: System architecture

From the Snort IDS generated alert file, the records of potentially bad traffic is cut and placed in a separate CSV file using the following script:

```
Grep -oP '/classification:\K.*' snort.txt>snort_cls.txt
Grep -oP 'TCP:\K.*' snort.txt>snort_tcp.txt
Sed-i 's/\[priority:2]//g' snort_cls.txt
Sed-i 's/\]//g' snort_cls.txt
Sed-i 's/DF//g snort_tcp.txt
Sed's/ \+/,/g' snort_cls.txt>snort_cls.csv
Sed's/ \+/,/g' snort_tcp.txt>snort_tcp.csv
Paste snort_cls.csv snort_tcp.csv>snort_fin.csv
```

The CSV file is uploaded on DataHero to perform analysis. Once the setup is complete we proceed to the DataHero dashboard by logging onto www.datahero.com/.

Figure 1 explains the network architecture of the proposed model. Two virtual machines are hosted on the (AWS) cloud environment. The VM running the wordpress server hosts the honeypots and the front end of the website.

The second virtual machine runs the MHN server which allows for easy provision and management of honeypots. Dionaea aims to trap malware exploiting vulnerabilities exposed by services offered over a network and ultimately obtain a copy of the malware. Snort and pOf are used for intrusion detection. Snort is a packet snifing tool and Pof is used for OS fingerprinting.

Figure 2 explains the system architecture of the proposed system. The MHN server and the honeynets are classified into different VPCs so that the attacker doesn't gain entry into the MHN server. A VPC is a virtual network that is logically isolated from other virtual networks in the AWS cloud.

RESULTS AND DISCUSSION

Analysis such as the countries with the most active botnets/attackers can be identified. Additional information such as the most attacked ports and the kind of malware used gives further insights about the kind of protection needed.

Information such as date, country, source IP (of the attacker), destination port number, protocol and honeypot are displayed in the attack report. The honeymap is used to track attacks as they happen in real-time. The above screenshot shows an attack from Latvia on the red dot (Fig. 3-7).

The data captured by Snort needs to be cleaned and converted from PCAP to a CSV format for it to be analyzed by DataHero. A Python script is used to automate this process. Figure 6 shows the proportion of potetially bad traffic and the kind of attacks used in the attempted information leak category.

A longer datagram length would indicate that the attacker is trying to flood the server with retransmission requests as the server retransmits if the received datagram packet is greater than the MTU (Maximum Transmission Unit).

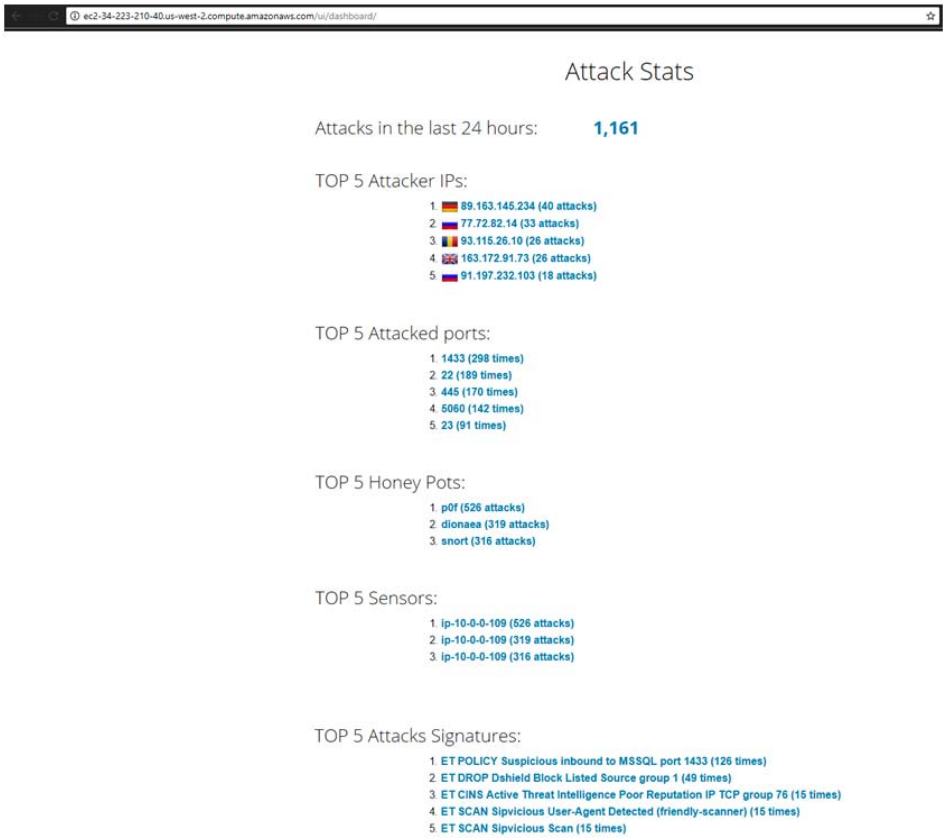

Fig. 3: A screenshot showing the top 5 attack statistics on the website

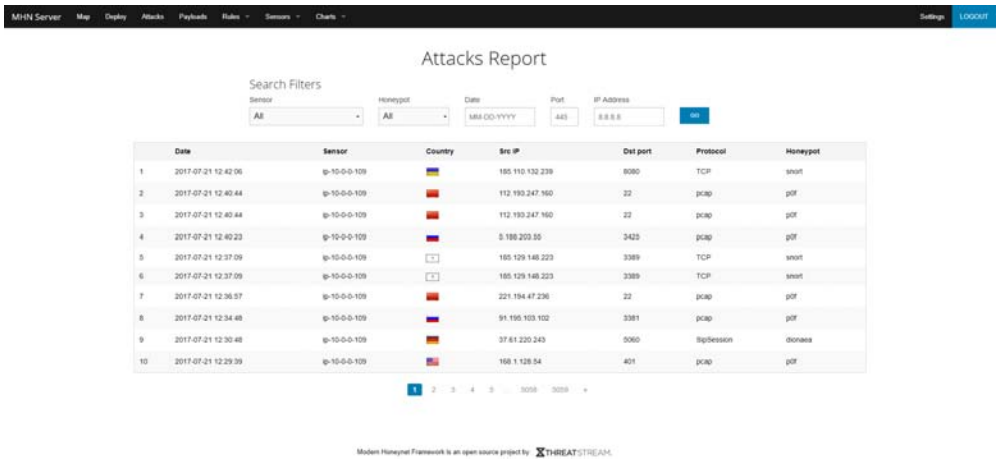

Fig. 4: A screenshot showing the attack report

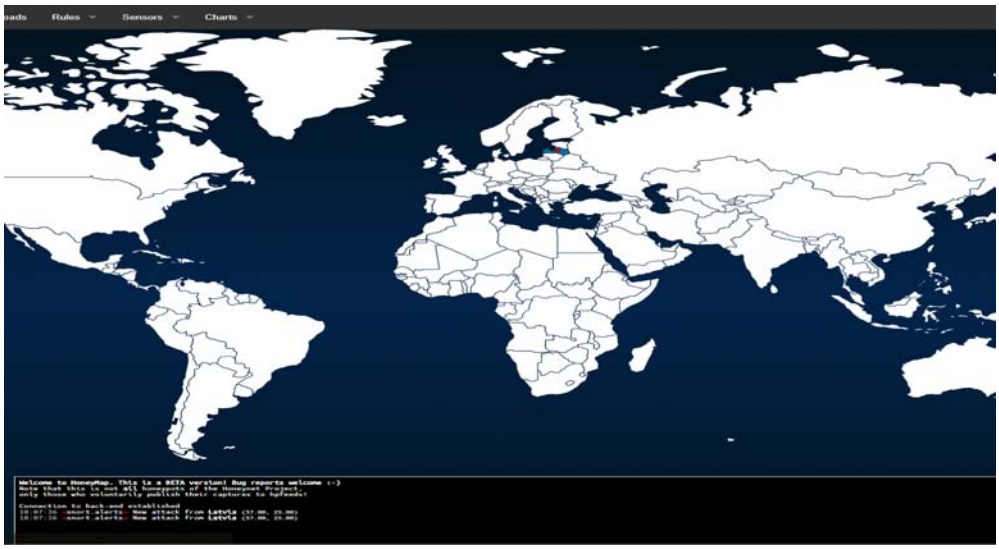

Fig. 5: A screenshot showing the honeymap

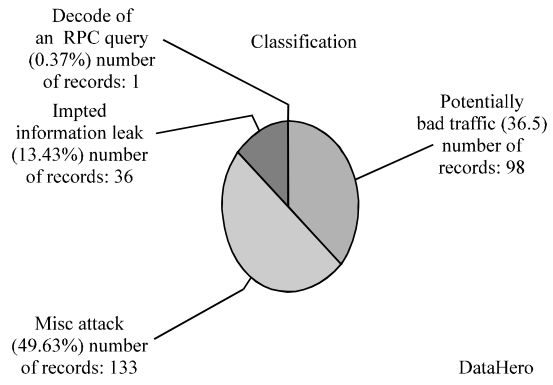

Fig. 6: A screenshot showing the classification of traffic as "potentially bad" from the captured Snort logs

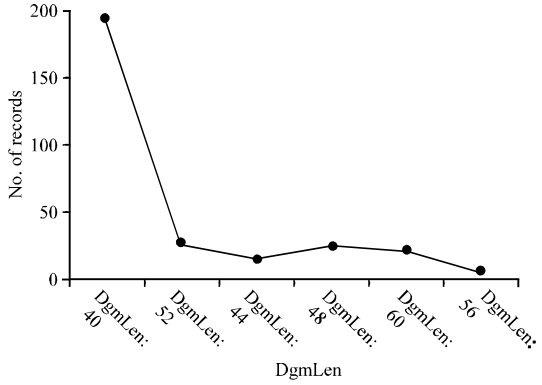

Fig. 7: A screenshot showing the datagram length of captured traffic from the Snort logs

CONCLUSION

Honeypots are effective tools for gathering data on what kind of malware and attack methods are used by malicious users. Honeypots on the cloud allow websites to have that extra layer of security which is essential to guard your data from hackers. By analyzing and accumulating information on attacks we can keep our system safe from attackers by hardening those areas. Honeypots have a wide range of applications from malware isolation to spam detection and even catching a rogue employee. Honeypots will see wider deployment as network security around the world improves and will help us make the internet a little bit safer.

IMPLEMENTATIONS

Implementing honeypots on a cloud environment is advantageous as it helps in minimizing setup and operational expenditure by utilizing virtual resources through the cloud provider. With the recent increase in cyber attacks and the sophistication of the malware that is available today due to government agencies has made cyber-security one of the most important fields of the coming decade. Honeypots will be seeing wider deployment and better technologies to implement them.

REFERENCES

Amoroso, E., 2010. Cyber Attacks: Protecting National Infrastructure. Butterworth Heinemann, Oxford, England, ISBN:978-0-12-384917-5, Pages: 233.

Chawda, K. and A.D. Patel, 2014. Dynamic and hybrid honey pot model for scalable network monitoring. Proceedings of the IEEE International Conference on Information Communication and Embedded Systems (ICICES), February 27-28, 2014, IEEE, Chennai, India, ISBN:978-1-4799-3698-4, pp: 1-5.

Dittrich, D., 2004. Creating and managing distributed honeynets using honeywalls. Master Thesis, University of Washington, Seattle, Washington.

Olagunju, A.O. and F. Samu, 2016. In search of effective honeypot and honeynet systems for real-time intrusion detection and prevention. Proceedings of the 5th Annual ACM Conference on Research in Information Technology, September 28-October 01, 2016, ACM, Boston, Massachusetts, ISBN:978-1-4503-4453-1, pp: 41-46.

PWHC., 2014. The global state of information security survey 2012. PricewaterhouseCoopers, London, England.

Polska, C., 2012. Proactive detection of security incidents: Honeypots. ENISA, Heraklion, Greece.

Spitzner, L., 2002. Honeypots: Tracking Hackers. Addison-Wesley, Boston, Massachusetts,.

Trost, J., 2017. Modern honey network via threatstream. GitHub Inc, San Francisco, California. https://github.com/threatstream/mhn.