Mitigating Computer Attacks in a Corporate Network using Honeypots: A Case Study of Ghana Education Service

Promise R. Agbedanu
Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

J. B. Hayfron-Acquah
Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

F. Twum
Department of Computer Science, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana

ABSTRACT
Computer and network security is increasingly becoming not only more significant to industry players but also complex regarding mitigating sophisticated cyber-attacks. It is essential for developers, systems administrators, and web administrators to develop and manage systems that can stand the test of time as far as computer and network attacks are concerned. A hybrid honeypot was deployed in the network setup of the Ghana Education Service. The honeypot set up was made up of Valhalla honeypot and honeyd (low-interaction honeypots), Cowrie (medium-interaction honeypot), Windows and two Ubuntu OS implemented on real systems (high-interaction honeypot) and Snort. This research goes a step further to collect the attack on data and analyse them. The attacks that were launched against the honeypots deployed in the network were Port Scanning, SSH Brute Force attack, HTTP Authentication Brute Force attack, SQL Injection and Spam. It was discovered that the honeypots received 5061 attack connections from October to December 2017. Majority of the attack connections were TCP based, resulting in 2851 of the total attack connections. The results of this work also show that honeyd receive 36% of the total attacks launched against the honeypots.

Keywords
Computer security, Network Security, Honeypots.

1. INTRODUCTION
Modern organizations such schools, banks, insurance companies and the security services, just to mention a few depend heavily on computers and its related applications to run their day to day services. In most of these organizations, computers are connected to form a network

With the growing trend in network and cloud computing, security has become the major concern of every organization that wants system availability, integrity and confidentiality.

2. BACKGROUND
On January 20, 2015, a Turkish was able to take down the official website of the Government of Ghana. Prior to this attack, the website of the Foreign Affairs Ministry was also hacked. During this attack nine other state agencies were also affected [1]. In that same year websites belonging to the Presbyterian University College, the University of Cape Coast, Kwame Nkrumah University of Science and Technology were also hacked [2].

3. RELATED LITERATURE
According to Lihet and Dadarlat [3], a honeypot is a fraudulent system that is deployed in a production environment to emulate a real system. Data found on a honeypot are not real, so when the honeypot is breached, it does not affect the actual network infrastructure. Honeypots can be grouped based on the level of interaction they provide, how they are implemented or where they are located.

Using honeypots as decoys to collect attack data can serve as a countermeasure against malicious threats in web applications. The use of honeypots that emulate web-based services and applications can help in collecting malicious activities by attackers. This work proposes a model using honeypots that were deployed and evaluated in different web environments [4].

According to Zhai and Wang [5], the use of honeypots in a campus network may end up expanding the network space which in turn may serve as a delusionary mechanism thereby prevent attacks by delaying or distracting attackers.

Kumar et al [6] proposed an integrated system that includes client and server honeypots. The server and the client honeypots are controlled by an active controller, which is a single centralized server. Their proposed systems have five functional components. These are, the client honeypots, server honeypots, honeypot controller, management and analysis server. The proposed framework utilizes honeypots to collect and analyse malware.

It is difficult to immediately generate detection rules based on the information gathered from honeypots. This work presented an agent-based honeypot framework to help remove malicious activities and executable files on servers infected by a zero-day attacks right after the honey detects such attacks [7].

One of the most popular form of attack deployed against mobile devices are malwares. Deploying honeypots like “honeypot-to-go” as a basic low-interaction honeypot to detect malwares is one of the comprehensive ways to mitigate malware attacks against mobile devices [8].

In order to prevent SQL injections, a model implemented from Snort and a honeypot has been proposed to curb this kind of attack. Though there are several solutions to SQL injection attack like hashing, query transmission and header sanitization but they all have some drawbacks. IDS rules are not updated automatically. However, the proposed solution

solves the problem by sending attack request to a proxy server which houses Snort IDS. The IDS make a decision as to whether to transfer the attack request to a real database server or to drop the request when it realizes that the attack signature matches an existing one. But, the real database converts login credentials entered to ASCII values and then matches the ASCII value to an already stored ASCII value in the database server. If the ASCII value matches with an existing one, the user is granted access. Otherwise, it is assume that an attacker is trying to compromise the system, so attacker is sent to the honeypot [9].

In this paper, an automated malware analysis framework is integrated with honeypot systems and Taiwan Malware Analysis Net (TWMAN) to simultaneously collect and analyse malware [10].

Zhan et al [11] proposed a framework that can be used to analyse attack data collected from honeypots. This framework which is the first statistical framework to analyse a honeypot's log file. This framework was used to analyse dataset from a low-interaction honeypot. However, the framework can equally be used to analyse dataset from high-interaction honeypots.

According to Leonard et al [12] securing information or data that is exchanged in a Body Area Network (BAN) especially from unauthorized people is very important to achieve confidentiality and integrity. Though solutions like cryptography can be implemented to ensure information security, they involve considerable performance in terms of overhead. To achieve both maximum security and minimal overhead, Leonard et al. proposed "wearable honeypot system." This Solution communicates false user health information between the base station and a set of some designated decoy node in the BAN. If the traffic is altered regarding content or arrival time, then it is flagged as an attack.

In a client-server environment, protocols such RDP and VNC are used in managing systems remotely. These protocols can be subjected to dangerous traffic from an attacker. According to Danchenko et al [13], the use of honeypots to emulate remote desktop connections can help administrators to collect attack data, analyse them and used the knowledge and the understanding gained from that to mitigate such attacks.

4. METHODOLOGY

Data was collected by deploying low, medium and high interaction honeypots in our network. These honeypots were strategically deployed in the network to help us understand how attacks are launched against our network. Data was gathered from the honeypots deployed in our network. The honeypots were configured to log all activities and attacks. The logs were then sent to a remote management machine in our network that has the necessary tools to process and statistically analysed the data gathered. The data gathered was then analysed to understand the various attacks that was launched against our honeypots, how the attacks were carried out, which vulnerabilities were exploited and the source of these attacks. For this work, a hybrid honey architecture is implemented. Figure 1 shows the honeypot architecture that was implemented.

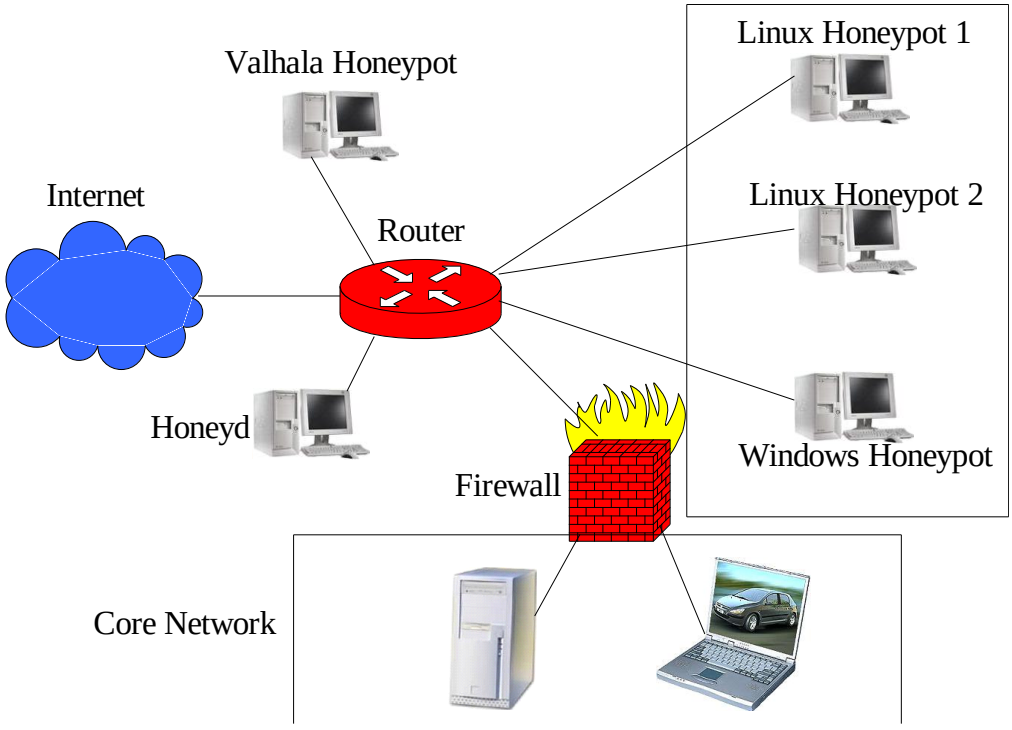

Figure 1- Implemented Honeypot Architecture

5. IMPLEMENTATION AND RESULTS

The honeypots were deployed and monitored for 90 days, from October to December, 2017. During the period of deployment and monitoring, over 5,000 identified attack connections were received. Table 1 shows the various tools and applications used in the honeypot setup.

Table 1. Tools for our honeypot implementation

Tools	Description	Specifications
Ubuntu 8	Physical Machine: High – Interaction Honeypot	Hp 6200 Pro Mini Tower Intel Core i7 Processor Speed: 3.4GHz 16GB RAM 1TB Storage
Windows Server 2003	Physical Machine: High – Interaction Honeypot	Hp 6200 Pro Mini Tower Intel Core i7 Processor Speed: 3.4GHz 16GB RAM 1TB Storage
Ubuntu 16.04	Virtual Machine: High – Interaction Honeypot	2GB RAM 40GB Storage
Valhala Honeypot	A low-interaction honeypot for windows	Version 1.9
Honeyd	A low-interaction honeypot for windows	
VMware	Virtualization Software	VMware Workstation 11.0.0 for Windows
Dionaea	Malware collector	
HoneyDrive	Open Source Linux Honeypot	HoneyDrive 3 (OVA)
Cowrie	Medium-interaction SSH honeypot	
ADHD	Active Defence Preinstalled tools	ADHD Version: **0.7.3**
Snort	Intrusion Detection System and Intrusion Prevention System.	Snort 2.9.8.2

5.1 Attacks Launched Against Honeypots

Most of the attacks focused on the low-interaction honeypots deployed in the network. Majority of the malicious activities deployed against these honeypots were port scans of different shades. Aside port scanning, it was observed that attacks like SSH Brute Force Attack, HTTP authentication brute force attack, SQL injection and spamming. Most of the attacks launched against our honeypots were done multiple times.

5.2 Statistical Analysis

This section shows the overall statistical analysis of the results collected from our honeypot during the period of our study. A total of 5061 attacks to our honeypots was received. Table 2 shows the number of attacks per protocol.

Table 2. Attack connections per protocol

Protocol	Connections	Percentage
TCP	2851	56.33%
ICMP	1442	28.49%
UDP	768	15.17%
Total	5061	100%

From table 2, it can be observed that majority of the attacks launched against our honeypots were TCP connection based with a total of 2851 attacks representing 56.33 percent. ICMP and UDP followed with 1442 and 768 attacks representing 28.49 percent and 15.17 percent respectively.

We also analyzed which TCP ports were mostly scanned and it turns out that port 22 was the most scanned port. From figure 2, port 22 received 968 scans, followed by port 21 with 675 scans and port 110 receiving the least number of scans. Figure 2 shows the top five TCP ports scanned.

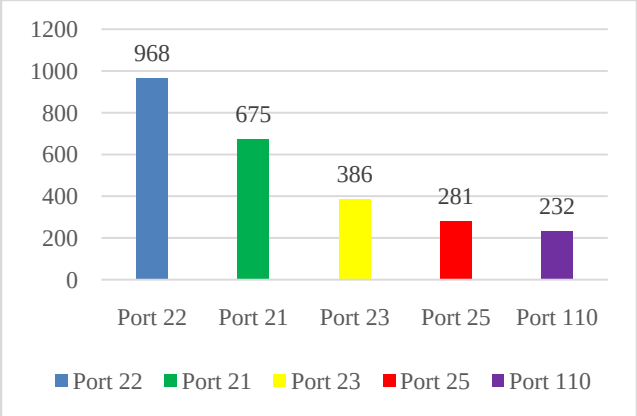

Figure 2: Top 5 scanned TCP ports

Fig 3 shows the number of attack connections per honeypot.

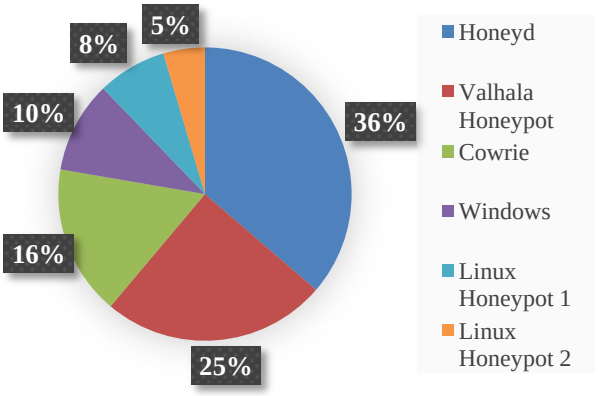

Figure 3: Number of attacks per honeypot

From the figure 3, the honeyd honeypot received 1838 attacks representing 36 percent, valhala honeypot received 1256 attacks representing 25 percent of the attacks launched against the honeypots deployed in the network. Cowrie, which was deployed as the medium-interaction honeypot received 838 attacks representing 16 percent whiles Windows honeypot, Linux honeypot 1 and Linux honeypot 2 received 511, 386 and 232 attacks respectively.

Although 823 unique IP addresses were observed coming from 18 countries all over the world, Russia dominated the list of countries that scanned the honeypots. From figure 4, the majority of the attacks launched against our honeypots were from Russia with 238 unique IP addresses followed by China with 165 unique IP addresses, the USA with 123 unique IP addresses, Morocco with 56 unique IP addresses attacks and Pakistan with 45 unique IP addresses.

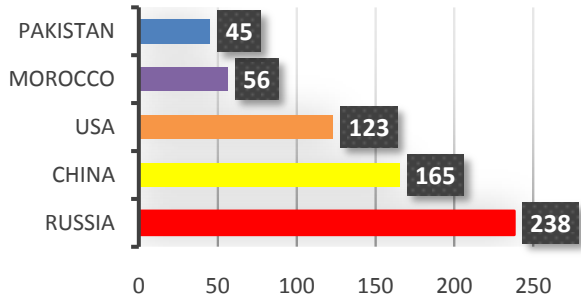

Figure 4: Top 5 countries with highest number of unique IP addresses

Considering attacks by type and how often they happened it was observed that Port Scanning was on top of the list of attacks launched against the honeypots. From figure 5, 31 percent of the attacks launched against the honeypots were in the form of Port Scanning (PS), followed by SSH Brute Force (SBF), Spam, HTTP Authentication Brute Force (HABF) and SQL Injection (SQLi) with 23, 18, 15 and 13 percent respectively.

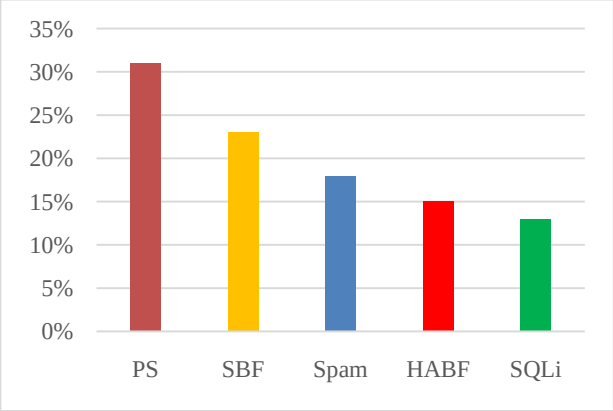

Figure 5: Type and frequency of attack

From the analysis it was observed that, the most username used in terms of frequency is admin, which had 467 login attempts, followed by root, mysql, guest and test with 456, 367, 321 and 245 login attempts respectively. This is represented in table 4.

Table 3. The 5 most attempted usernames

Username	Number of login attempts
Admin	467
Root	456
Mysql	367
Guest	321
Test	245

From figure 6, it is also clear that the most attempted password is admin with 511 login attempts followed by root with 442 login attempts. The rest are password with 317 login attempts, 123456 with 217 login attempts and test with 183 login attempts.

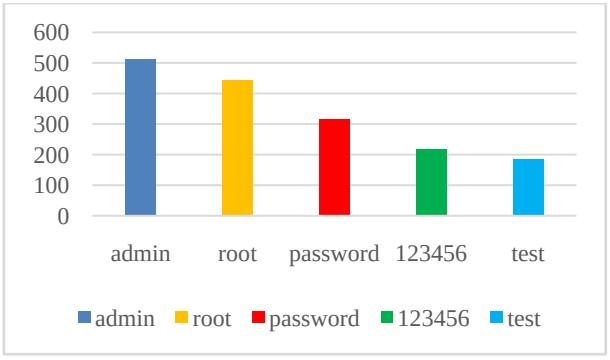

Figure 6. Top 5 attempted passwords

The top five username and password combinations and how many times attackers tried to use such combinations to breach the honeypots deployed in the network were analysed. From the analysis, the most username and password combination used is admin and admin respectively. This username and password combination was 256 times against our honeypots. This was followed by root and root which was used 232 times with root and password, root and 123456, test and test also following in an orderly manner with attempts of 211, 187 and 167 respectively. This is represented in table 5.

Table 4. Most 5 tested usernames and passwords

Username	Password	Number of attempts
Admin	Admin	256
Root	Root	232
Root	Password	211
Root	123456	187
Test	Test	167

6. CONCLUSION AND FUTURE WORK

In this work, a hybrid honeypot is deployed in a corporate network using the Ghana Education Service as a case study. The honeypot was deployed in the network for 90 days, that is from October to December 2017. The attacks data gathered by the honeypots were analysed. The findings were then used to strengthen the core network of the organization. The deployment of honeypots has proven to be an excellent mitigation strategy for most forms of attacks because a well-implemented honeypot ends up wasting the time of an attacker. In most instances, the attacker may leave the network thinking that he actually breached a real system giving the network administrator an opportunity to patch up the vulnerability the attacker exploited in the honeypot environment. The deployment of honeypots in GES and combining with other existing security infrastructure improved the system and network security of this particular institution.

Based on our results from the attack logs, the following as recommendations for Small Office Home Office (SOHO) and corporate network environment are proposed. To being with, systems administrators should never use the default usernames and passwords of systems they deploy in their

network. Moreover, administrators should always use strong passwords and make sure the systems they deploy have up to date security patches. To strengthen the security of networks, honeypots should periodically be deployed in such networks to find out what kind of attacks are being launched against real systems in the network and what kind of vulnerabilities are they exploiting. Future work will involve the use of the attack data from this work to learn more about attacker's skills by replaying the attacks launched against the honeypot network to really understand how these attacks were deployed.

7. REFERENCES

[1] M. Ansah, "Gov't of Ghana website hacked | citifmonline," *citifmonline*. 2015.

[2] NA, "KNUST's Official Website Hacked!.," *233 Live News*, 2015. [Online]. Available: https://233livenews.wordpress.com/2015/11/24/knusts-official-website-hacked/. [Accessed: 20-Apr-2016].

[3] M. A. Lihet and V. Dadarlat, "How to build a honeypot System in the cloud," in *2015 14th RoEduNet International Conference - Networking in Education and Research (RoEduNet NER)*, 2015, pp. 190–194.

[4] N. Kuze, S. Ishikura, T. Yagi, D. Chiba, and M. Murata, "Detection of vulnerability scanning using features of collective accesses based on information collected from multiple honeypots," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 1067–1072.

[5] J. Zhai and K. Wang, "Research on applications of honeypot in Campus Network security," in *Proceedings of 2012 International Conference on Measurement, Information and Control*, 2012, vol. 1, pp. 309–313.

[6] S. Kumar, R. Sehgal, and J. S. Bhatia, "Hybrid honeypot framework for malware collection and analysis," in *2012 IEEE 7th International Conference on Industrial and Information Systems (ICIIS)*, 2012, pp. 1–5.

[7] I. S. Kim and M. H. Kim, "Agent-based honeynet framework for protecting servers in campus networks," *IET Inf. Secur.*, vol. 6, no. 3, pp. 202–211, 2012.

[8] W. Z. A. Zakaria, F. M. Maksom, and K. Abdullah, "Observing the presence of mobile malwares using low-interaction honeypot," in *2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2016, pp. 117–121.

[9] U. Upadhyay and G. Khilari, "SQL injection avoidance for protected database with ASCII using SNORT and HONEYPOT," in *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2016, pp. 596–599.

[10] Y. L. Tsai, B. Y. Lee, and J. G. Chang, "Automated Malware Analysis Framework with Honeynet Technology in Taiwan Campuses," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, 2012, pp. 724–725.

[11] Z. Zhan, M. Xu, and S. Xu, "Characterizing Honeypot-Captured Cyber Attacks: Statistical Framework and Case Study," *IEEE Trans. Inf. Forensics Secur.*, vol. 8, no. 11, pp. 1775–1789, 2013.

[12] A. M. Leonard, H. Cai, K. K. Venkatasubramanian, M. Ali, and T. Eisenbarth, "A honeypot system for wearable networks," in *2016 IEEE 37th Sarnoff Symposium*, 2016, pp. 199–201.

[13] N. M. Danchenko, A. O. Prokofiev, and D. S. Silnov, "Detecting suspicious activity on remote desktop protocols using Honeypot system," in *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2017, pp. 127–128.