Analysis and mitigate SSH Brute-force and Dictionary attacks using honeypot system

Muneer Alwazzeh[(1)] Dr. Sameer Karaman, Dr. Mohammad Nour Shamma

Abstract

Computer networks are vulnerable to cyber-attack, which has been increased rapidly and caused harm to our network security systems. It is necessary to build a system with the ability to deceive, detect and block these attacks. SSH Brute-force and dictionary attacks are ones of the most famous attacks on internet and computer networks. Honeypots are effective security system to analyze and mitigate SSH brute-force and dictionary attacks. Cowrie honeypot system was deployed to both records and analyzes SSH Brute-force and dictionary attack information and command execution after successful login to Cowrie system. Evaluation of Cowrie honeypot system can be obtained by using confusion matrix and accuracy with high percentage, which means that Cowrie honeypot system, has a good ability to protect our network from SSH attacks.

Keywords: SSH attacks; Medium interaction honeypots; Cowrie honeypots

[(1)] Faculty of electrical and mechanical engineering, Damascus University, Damascus, Syrian Arab Republic

[*] Corresponding author, malwazzeh@aec.org.sy

تحليل وتخفيف هجمات القاموس والقوة الغاشمة على بروتوكول SSH باستخدام نظام مصائد مخترقي الشبكات

منير الوزة [1] د. سمير كرمان د. محمد نور شمه

الملخص

تكون شبكات الكمبيوتر عرضة للهجمات الإلكترونية، والتي زادت بسرعة وتسببت في إلحاق الضرر بأنظمة أمان الشبكة الخاصة بنا. من الضروري بناء نظام لديه القدرة على خداع هذه الهجمات واكتشافها وصدها. تعد هجمات القوة الغاشمة والقاموس على بروتوكول SSH من أشهر الهجمات على شبكات الإنترنت والكمبيوتر. مصائد مخترقي الشبكات هي نظام أمان فعال لتحليل وتخفيف هجمات القوة الغاشمة والقاموس على بروتوكول SSH. تم نشر نظام مصائد مخترقي الشبكات Cowrie لتسجيل وتحليل المعلومات والأوامر المنفذة لهجمات القوة الغاشمة والقاموس على بروتوكول SSH وذلك بعد تسجيل الدخول بنجاح إلى نظام Cowrie. تم الحصول على تقييم نظام مصائد مخترقي الشبكات Cowrie باستخدام مصفوفة الارتباك والدقة وبنسبة عالية، مما يعني أن نظام مصائد مخترقي الشبكات Cowrie لديه قدرة جيدة على حماية شبكتنا من هجمات SSH.

الكلمات المفتاحية: هجماتSSH ، مصائد مخترقي الشبكات متوسطة التفاعل، مصائد مخترقي الشبكات Cowrie

[1]كلية الهندسة الكهربائية والميكانيكية، جامعة دمشق، دمشق، الجمهورية العربية السورية
المؤلف المراسل malwazzeh@aec.org.sy

1. Introduction

Internet and computer networks are needed in science, technology and other areas of life. Their threats and vulnerabilities are increasing rapidly, causing loses of important data. Improving abilities of network security techniques is highly demanded to protect data from intelligent hackers [1]. One of the famous attacks on internet and computer networks is SSH brute-force and dictionary attack that aims to break authentication of system by using of combination of user names and passwords.

RFC4251 defined Secure Shell, as "SSH", is a protocol for secure remote login and other secure network services over an insecure network"[2]. SSH provides encrypted communication, password-less login via public key authentication, and host-based verification. SSH listens on the standard Transmission Control Protocol (TCP) port 22. In Secure Shell session, the client first establishes TCP connection with the SSH server and then exchanges authentication information. Then the client sends secure shell login request. Username and password combination will be checked by SSH server, which decides whether the client is authorized, or not. In SSH dictionary attacks, the hacker uses a single attacking machine to commence dictionary attacks targeted to multiple SSH servers having different destination IP addresses, or large set of botnets having different IP addresses that are used to attack a single victim SSH server[3]

Honeypots are assistant network security technique are defined, according to Lance Spitzher, the inventor of the honeypot idea as: "a source of information whose value depends on the unverified or prohibited use of that source"[4]. Honeypots are built to lure and study the tactics of hackers and their intents and to improve the security policies. Honeypot is a system or program that is introduced to network to have the system probed, attacked, and potentially exploited.

Honeypots are classified according to several factors:

Relying on interacting with hackers:

- **Low-Interaction Honeypot:** It is issued to detect and deceive attackers by simulating Operating System services and gateway services on the Operating System host. The interaction is limited and the honeypot does not have its own operating system, and the primary mission is to slow down the attack.

- **Medium Interaction Honeypot:** It has the same principle as the Low Interaction Honeypot. It provides the attacker with phishing by having an operating system, as the attacker communicates with a large number of simulated services.

- **High Interaction Honeypots:** Includes real operating systems and applications like a real FTP server. It has the greatest threat as it exposes itself to the attacker for an extended period of time. They should be kept under constant surveillance due to their security risks[4].

Depending on the purpose or goal:

- **Productive Honeypots:** are systems that help mitigate the risk to the organization, and are placed close to the servers in the network. They are intended to simulate real production systems so that the attacker will spend time and resources attacking them.

- **Research Honeypots:** are mainly used to uncover information about new methods of attacks, viruses, and worms. They are difficult to maintain and complex in structure and provide information on the Black hats and their offensive policies[5].

Honeypot configure to log all activities and attacks. The gathered data was then analyzed to understand SSH brute-force and dictionary attacks, how the attack was carried out, which vulnerabilities were exploited and sources of the attacks and their IP.

In our paper, we deployed Cowrie honeypot system to collect username and password combination that are attempted by

SSH brute-force and dictionary attack targeting secure shell service. We have also recorded all commands and keystroke they executed by attackers after successful logins to Cowrie honeypot system.

The rest of this paper is organized as follow: in section 2, we introduced the related works on SSH attacks. Section 3 the honeypot system configuration, programs and facilities for deployment of the proposed model were discussed. In section 4 we presented the deployed honeypot experimental results and discussed various connection attempts made to the honeypot along with the SSH brute- force attacks. Finally, section 5 introduced conclusion and future works.

2. Related work

Honeypots are active defense techniques on network security area. Network researchers have used honeypots to study SSH brute-force and dictionary attacks. Esmaeil et al. modified low interaction honeypot (KOjoney) that listened for attack traffic on TCP port 22 and recorded attempts to gain remote access to honeypot and suggested some recommendations to protect from SSH attacks[6]. Konaiaris et al. deployed their honeypot system as a web trap for attacker who try to gain illegal SSH service access. They proved that honeypots are effective tools in gathering information about SSH attacks. Additionally, they introduce a visualization tool to help security researchers during analysis and conclusions drawing phases[7].Solomon et al. deployed Kippo honeypot to analyze secure shell and dictionary attack. They recorded all passwords, usernames and IPs of attacker machine. They also collected commands of successful logins[8]. Praful et al. utilized “Conch”, which is an SSHv2 implementation written in Python open service at port 22. When attacker tries to scan this port for information, data is logged with IP address of attacker, all attempt of access and commands are recorded and then data is analyzed, and malicious traits are discovered[9]. Devi and Aulia presented Cowrie honeypot system for SSH server service and visualize collected information through the Kippo-graph programme to save SSH server from brute-force attacks and monitor server and analyze the behavior of the attacker[10].

Promise et.al deployed hybrid honeypot in network of Ghana Education Service to record and analyze different types of attacks. They proved that honeypots are excellent mitigation strategy for most forms of attacks and are efficient to waste time of attacker[11].

The previous mentioned work above showed the importance of network monitoring and the necessity of attack prevention for SSH brute-force and dictionary attacks and striving to build a robust detection, monitoring and analysis system capable of stopping such type of attacks with high efficiency.

3. proposed system:

SSH protocol attacks became very familiar type of attacks on information security. There are few types of SSH attacks such as SSH port scanning, SSH penetration, and SSH brute-force and dictionary attacks. In Brute-force attack, attacker try to use all possible character of password by using dictionary, and large lists of password. Moreover, servers limit the number of authentication attempts per connection, but attacker may bypass them, performing SSH brute-force and dictionary attack[7].

This research aims to design and set up SSH honeypot system to study SSH brute-force, dictionary attacks and intrusion activities and inhibit them. For this purpose, we deployed cowrie honeypot as medium interaction honeypot to study SSH based attacks and collect connection attempts on port 22.

Cowrie is an SSH and Telnet honeypot with medium interaction level used to record brute- force attack and SSH requests. It provides shell environment to interact with attacker and record all commands executed after the successful username and password prediction of the dictionary attack[12].

Cowrie has disadvantage that it displays data in form of log system, and it is

inefficient for network administrators to monitor logs and analysis attacks[10].

Cowrie is distributed honeypot logs SSH interactions in MYSQL database[13].

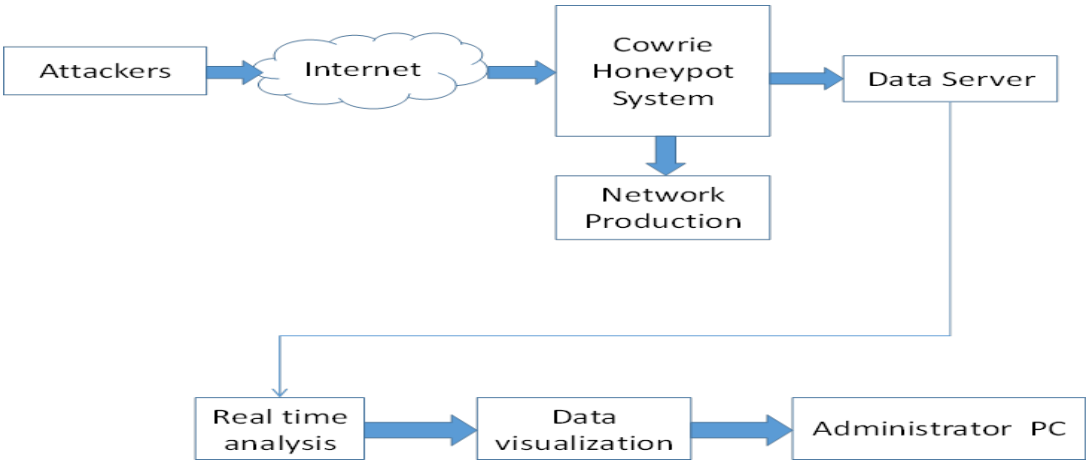

Figure (1) proposal design

Our proposal design depend on three section: detection, defense and alert sections.

Detection section is the most important section in this design. It is responsible for detecting any malicious activity in our design.

Defense section protects the network and blocks the SSH attack before damaging the network.

Alert section report all details about attacks including: type of SSH attack, number of login attempts, attacker IP source, username and password, and commands and keystrokes.

The process of detection, defense and alert are carried out simultaneously in the proposed model according to the following steps:

The attacker will be directed to a honeypot system to interact with. The Cowrie honeypot detects any malicious activity, and all peripheral activity such as commands and keystrokes, which will be logged in log files until the attacker logs out of the system.

The attacker will first enter the username and password, and the system will check it and determine if it is a legitimate user or not depending on the number of login attempts to the system, the number of TCP packets flowing, and the source of the attack. For example, if the attacker tries more than three times then he is an illegal user.

If the user is a legitimate user, the system will route him to the production network (the real network) or it is illegitimate and therefore the system will collect information about this attacker to determine the SSH attacks, which include username, password, number of login attempts, attacker IP, number and size of packets, and instructions and commands that the attacker executed and then will send the alert and attack details via email to the system administrator.

The collected data will be stored and then all data will be sent to a real-time analysis system for visualization. The administrator will decide whether to resume the connection (further interaction with the attacker) or cut it.

Figure2 show the flow chart of proposal design.

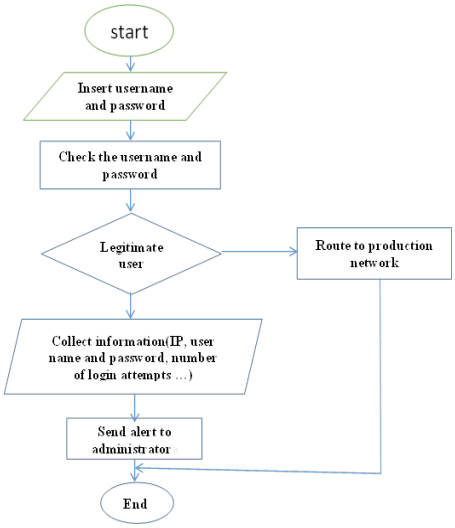

Figure (2) flow chart of proposal design

4. Discussion and results:

To implement our honeypot system, we setup it in virtual environment using VMware workstation. We configured Cowrie honeypot system on virtual Ubuntu 12.04 guest operating system. The hosting machine is also Ubuntu operating system. Cowrie honeypot system was ready to capture SSH attacks. Our Cowrie honeypot system had a public IP address so that it would be easily accessed from the Internet. Cowrie is configured to accept SSH connection on TCP port 22. It collected all activities of attacker including username, password, and commands run by attacker. We had also installed MySQL server on the system to store all database.

Brute-force and dictionary attacks:

We summaries the brute-force and dictionary attacks results collected from our Cowrie honeypot system. Cowrie honeypot system encounters about 8000 SSH connection requests in 30 days. 12092 login attempts coming from 467 Unique IP addresses.

A total of 467 distinct IP addresses connected to Cowrie honeypot system. Table 4 shows the top 10 countries where attackers were originating. The attacks were received from multiple countries, and the largest part was, as expected, from China, with 43.55% of the attacks, followed by the United States of America and then Russia. The attack was also carried out from countries that were not expected and not mentioned in previous results and studies, such as Morocco and Indonesia, and at reasonable rates. This indicates the spread of attackers and the phenomenon of penetration in most countries.

Table (4): top 10 countries with percentage

country	Login attempts	percentage
China	5266	43.55
USA	3014	24.93
Russia	1319	10.91
France	817	6.76
Korea	626	5.18
Pakistan	309	2.56
Morocco	299	2.48
Hungary	191	1.58
Indonesia	138	1.15
Philippine	99	0.82

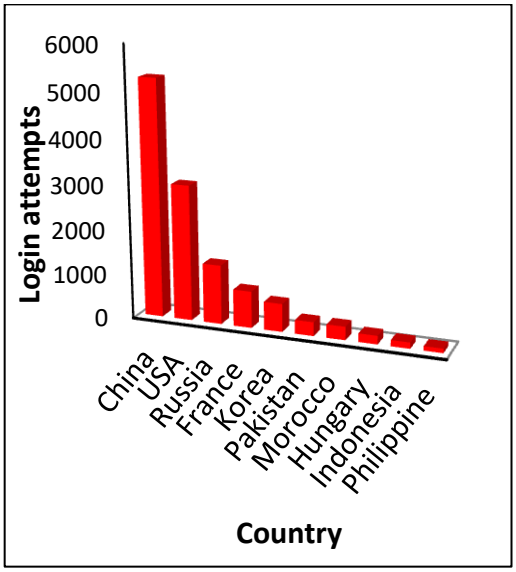

Figure (2) top 10 countries with percentage

Attackers gain access to Cowrie honeypot system with 233 distinct username. The most frequently username is root in 5896 times which is about 50 percent of the total login attempts. Table 5 shows the top 10 username with their login attempts percent.

Various usernames were used by the attackers, with "Root" being the highest percentage, followed by "Admin". It also used new words that were not mentioned in previous studies, such as "Ubuntu", "Ftp", and other random words. This indicates that the attackers did not adhere to dictionaries

and tried to use unexpected new words.

Table (5) top 10 username with percentage

Username	Number of attempts	percentage
root	5896	48.75
admin	1285	10.62
guest	953	7.88
test	398	3.29
user	255	2.11
support	212	1.75
mysql	174	1.43
ftp	150	1.24
ubuntu	123	1.01
temp	91	0.75

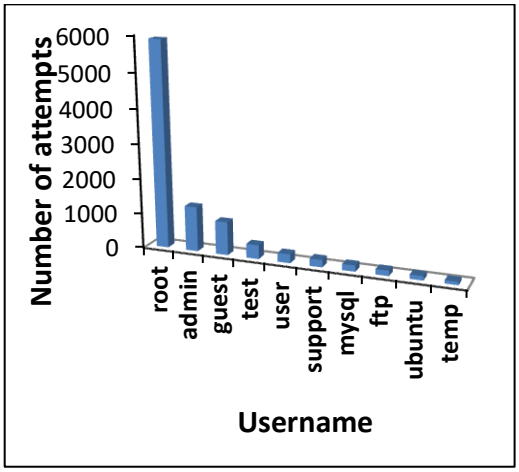

Figure (3) top 10 username with percentage

We have configured our Cowrie honeypot with familiar password that can be guessed by attackers. We have collected 4942 unique password. The most frequently attempted password is admin, which occurs 1174times. Table 6 shows top 10 passwords with their percent.

The higher percentage of password used by the attackers was "Admin" instead of "Root" as expected, and this indicates a change in the attackers' methods that they use and their lack of commitment to traditional methods, in addition to the emergence of strange passwords that did not appear in any of the dictionaries and previous studies.

Table (6) top 10 password with percentage

Password	Number of attempts	percentage
admin	1174	23.75
root	910	18.41
test	708	14.32
testtest	654	13.23
guest	405	8.19
123456	340	6.88
password	256	5.18
1234	113	2.28
user	98	1.98
ftp	27	0.54

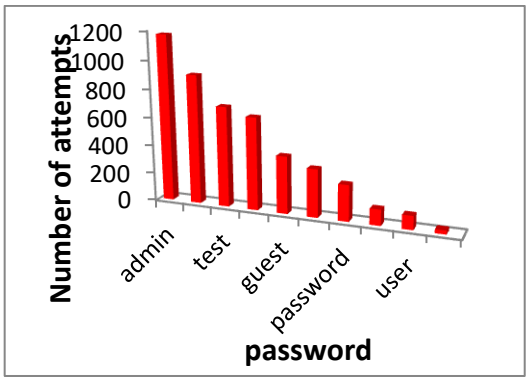

Figure (4) top 10 password with percentage

We have also observed the common username and password combinations as shown in table 7. The frequency of username and password pairs was as expected with the admin\admin pair being the most frequent in these results, followed by the root/root pair.

Table (7) top 10 username/password with percentage

Username/password	Number of attempts	percentage
admin/admin	511	21.32
root/root	400	16.69
admin/root	335	13.98
admin/123456	291	12.14
root/123456	254	10.6
user/user	199	8.3
root/password	112	4.67
guest/guest	105	4.3
ftp/ftp	24	1.02
support/support	11	0.46

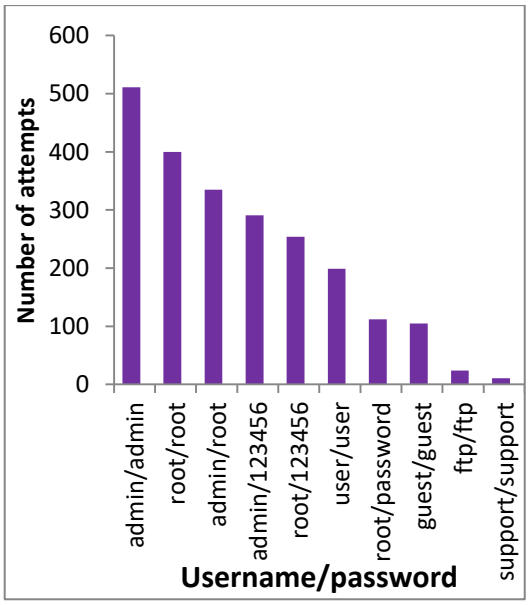

Figure (5): top 10 username/password with percentage

Attackers use various SSH client version to connect to Cowrie honeypot system. As shown in table 8, the most frequently used version is libssh2 version 1.4.3. libssh is a multiplatform C library implementing the SSHv2 and SSHv1 protocols and can remotely execute programs and transfer files[14].

The attackers used different versions of the SSH client to connect to the Cowrie honeypot. libssh2 version 1.4.3 is the most widely used and this indicates that attackers are keeping up to date with the latest versions and developing their methods of penetration into networks.

Table (8) top 10 attacker agent with percentage

Attacker agent	percentage
SSH-2.0-libssh2_1.4.3	75.7
SSH-2.0-jsch-0.1.51	8.3
SSH-2.0-Erlang/4.0	3.5
SSH-2.0-libssh2_1.4.2	2.1
SSH-2.0-libssh2_1.4.0	1.8
SSH-2.0-Putty_Local	1.4
SSH-2.0-libssh2_1.6.0	1.1
SSH-2.0-dropbear_0.47	1.1
SSH-2.0-Putty_release_0.63	0.8
SSH-2.0-MEDUSA_1.0	0.6

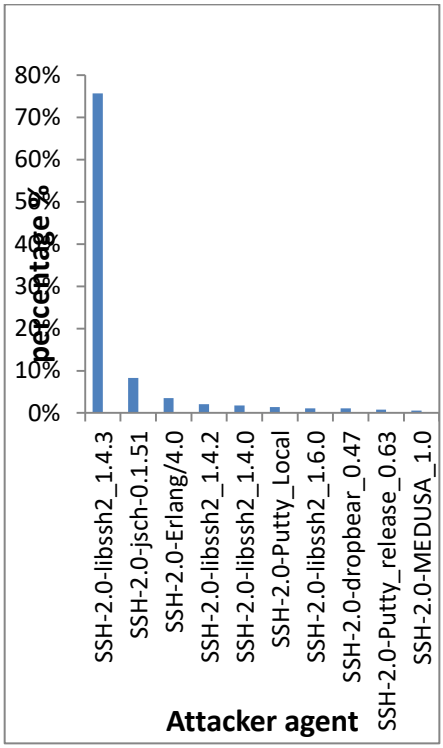

Figure (6) top 10 attacker agent with percentage

After attackers found the correct username and password using Brute-force and dictionary attacks, next step is to run Linux shell commands. The attacker tries to obtain information about the server such as the operating system used, the processes running on the system, and whether it can download malicious files and programs, or steal information from the server.

The command "W" has the largest percentage of used commands, which provides information about who has logged into the system and their login history.

The attackers also executed the "wget" command to download the files. These downloaded files are secretly copied to our honeypot directory for future analysis. The majority of the downloaded files were sh, and executable scripts.

We store these files in a separate place and all the information collected about the attacker simultaneously, so that we have a backup copy of these files and information (with the possibility of encrypting this information in order to protect it), because some attackers can remove traces of their presence in the system logs in the honeypot "var", which is clearly suspicious activity, and after attackers delete the records from the

system, they create a new file with a similar name. We can also later analyze files downloaded by the attackers and detect new malware that the attacker might spread.

The most frequently executed commands are shown in table 9

Table (9) top 10 command with percentage

Commands	percentage
w	45,55
uname	15.48
ls	12.22
ld	11.97
wget	6.17
passwd	5.01
history	1.22
help	1.18
Netstat	0.83
Ifconfig	0.51

System accuracy:

To measure the accuracy of Cowrie honeypot system, the degree of successfulness of system was determined by using confusion matrix and accuracy[15], which is described as following:

Table (10) Confusion Matrix

Prediction	Actual	
	T(cowrie)	F(production)
T(attacker)	TP(8000)	FP(5)
F(user)	TN(15)	FN(200)

True positive (TP) means the number of attackers, which successfully login to Cowrie system and interact with Cowrie honeypot.

True negative (TN) means the number of legitimate user, which successfully login to Cowrie system and interact with Cowrie honeypot.

False positive (FP) means the number of attackers, which successfully login to production network.

False negative (FN) means the number of original or legitimate user, which routed to the network production.

The threshold of success rate of experiment based on the subjectivity of the user should be beyond 80% to be considered as succeed.

$$\frac{(TP + TN)}{Total\ Data} \times 100 = \boldsymbol{Accuracy}$$

More than 8,200 SSH attacks were reported as the attackers interacted with the Cowrie honeypot system and about 8000 valid attacks were detected by the system and logged information about them, 200 users were correctly routed to the network production and marked as genuine users, and 15 users successfully login to Cowrie system and interact with Cowrie honeypot. The system failed to identify 5 attacks as true attacks.

Based on the data that was recorded in this research, the accuracy of system can be calculated as below:

TP=8000

TN=15

FP=5

FN=200

Accuracy= $\frac{15+8000}{8220} \times 100 = 97.7\%$

Calculation of accuracy showed that the rate of accuracy resulting from implementation of Cowrie honeypot system is more than 97%, and this indicates the efficiency of the proposed system and its ability to distinguish attacks significantly and with high accuracy. While we note that the accuracy of the proposed systems in previous studies was not discussed, and they were satisfied with presenting some of the results obtained, and the extent of the efficiency of the systems and their distinction for the incoming attacks were not specified.

Therefore, our Cowrie honeypot system succeeded in protecting network from SSH brute-force and dictionary attacks, and its ability to collect information and distinguish attacks significantly and with high accuracy and the possibility of its contribution to the detection of new attacks and malware.

5. Conclusion and future work:

In this research, Cowrie honeypot system was deployed in network for 30 days. All brute-force and dictionary SSH attacks data and execute commands after successful login to Cowrie honeypot system were gathered and analyzed. The deployment of Cowrie honeypot system has proven to be an effective mitigation technique for SSH attack with high accuracy percentage. To strength the security of our network, high interaction honeypot should be deployed to learn about attacker's skills and collect more information about the new types of attacks and mitigate intrusions.

Acknowledgements

Prof. Hasan Abou Alnoor has contributed to the work presented here but unfortunately passed away just before submitting it. Authors sincerely dedicate this work to the memory of Prof. Abou Alnoor.

References

[1] I.T. Plata, E.B. Panganiban, B.B. Bartolome, ASecurity Approach for File Management System using Data Encryption Standard (DES) algorithm, International Journal of Advanced Trends in Computer Science and Engineering, 8 (2019).

[2] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/info/rfc4251>.

[3] A. Satoh, Y. Nakamura, T. Ikenaga, A flow-based detection method for stealthy dictionary attacks against Secure Shell, Journal of Information Security and Applications, 21 (2015) 31-41.

[4] L. Spitzner, Honeypots: Tracking Hackers, Addison-Wesley Longman Publishing Co., Inc.2002.

[5] S.D. Lakshmi, G. Arunkumar, V.M. Viswanatham, Network Security Enhancement through Honeypot based Systems, International Journal of Engineering and Technology, 7 (2015) 290-293.

[6] E. Kheirkhah, S.M.P. Amin, H.A. Sistani, H. Acharya, An Experimental Study of SSH Attacks by using Honeypot Decoys, Indian Journal of Science and Technology, 6 (2013) 5567–5578.

[7] I. Koniaris, G. Papadimitriou, P. Nicopolitidis, Analysis and visualization of SSH attacks using honeypots, Eurocon 2013, 2013, pp. 65-72.

[8] Z. Solomon, P.S.A. Melese, Honeypot System for Attacks on SSH Protocol, International Journal of Computer Network and Information Security(IJCNIS), 8 (2016) 19-26.

[9] P. Nair, V. Nair, K. Nair, P.K.S. Charumathi, Implementation of Honeypot to Trap and Track Cyber Attacks, International Research Journal of Engineering and Technology (IRJET, 7 (2020) 970-974.

[10] D.A.P. Putri, A. Rachmawati, Honeypot Cowrie Implementation to Protect SSH Protocol in Ubuntu Server with Visualisation Using Kippo-Graph, International Journal of Advanced Trends in Computer Science and Engineering, 8 (2019) 3200-3207.

[11] R. Promise, J.B. Hayfron-Acquah, F. Twum, Mitigating Computer Attacks in a Corporate Network using Honeypots: A Case Study of Ghana Education Service, International Journal of Computer Applications, 180 (2018) 18-22.

[12] W. Cabral, C. Valli, L. Sikos, S. Wakeling, Review and Analysis of Cowrie Artefacts and Their Potential to be Used Deceptively, 2019 International Conference on Computational Science and Computational Intelligence (CSCI), 2019, pp. 166-171.

[13] S. Dowling, M. Schukat, E. Barrett, Improving adaptive honeypot functionality with efficient reinforcement learning parameters for automated malware, Journal of Cyber Security Technology, 2 (2018) 75 - 91.

[14] Libssh-The SSH library (2011), Available from https://www.libssh.org/

[15] S. Visa, B. Ramsay, A. Ralescu, E.v.d. Knaap, Confusion Matrix-based Feature Selection, in: S. Visa, A. Inoue, A.L. Ralescu (Eds.) Proceedings of The 22nd Midwest Artificial Intelligence and Cognitive Science Conference 2011, Cincinnati, Ohio, USA, April 16-17, 2011, CEUR-WS.org, Cincinnati, Ohio, USA, 2011.