Securing Network using Honeypots: A Comparative Study on Honeytrap and T-Pot

Sandip Biswa[1], Pema Wangmo[2], Tshoney Rangdel[3], Tashi Wangchuk[4*], Younten Tshering[5], and Tashi Yangchen[6]

[1-6] *Department of Information Technology, Jigme Namgyel Engineering College, Royal University of Bhutan*

[*] *Corresponding author: tashiwangchuk.jnec@rub.edu.bt*

Published: February 2025

Abstract

A wider attack surface has been produced by the growing reliance on technology and the internet for communication, business, and other activities, making it simpler for attackers to identify flaws and use them to their advantage. The sophistication of cybercriminals is also rising, and they are adopting cutting-edge methods and technologies to get beyond conventional security measures and avoid detection. With the implementation of honeypots, networks can defend against potential attacks and safeguard data on the systems and networks. A honeypot is a decoy system set up as part of a security mechanism to draw in potential attackers and record their activities for further examination. Two honeypot solutions, namely Honeytrap and T-Pot were deployed and implemented in the simulated private network to study the hardware requirements, installation complexities, range of emulated services supported, and the level of their interaction. Through analysis, it was found that both Honeytrap and T-Pot contribute to enhancing the systems and network security underscoring the significance of honeypots as valuable resources for security networks and systems.

Keywords— Cybersecurity, T-Pot, Honeytrap, Honeypot, cyberattack

1 Introduction

Organizations have to be very proactive to protect their systems and networks from the increasing cyber threats that are becoming more sophisticated. A honeypot is a security mechanism designed to detect, identify, and examine cyber-attacks by both system insiders and external penetrators to gain access to information systems. It essentially functions as a decoy system that mimics the characteristics of a real system to draw in and catch attackers. Typically, honeypots are deployed to detect and analyze new attack techniques, gather intelligence on attackers, and enhance system security by identifying vulnerabilities. Honeypots come in various forms; they can be low-interaction honeypots that simulate a few services or high-interaction honeypots that emulate an entire network.

They can be implemented inside or outside organization's network and can be customized to suit specific security needs. Honeypots can be a useful tool in the battle against cybercrime, providing important information about the behaviour of attackers and also assisting in the improvement of their security posture. This project explored the honeypot by implementing two distinct software solutions, Honeytrap and T-Pot, and assessed their suitability based on the minimum hardware requirements, installation complexities, range of services emulated, and the level of interaction supported by each.

2 Literature Review

2.1 Cybersecurity History

Today, people often forget what life was like before technology became so prevalent. Back then, people did not worry much about cybersecurity because connected devices were not widely used [1]. However, it's crucial to remember the past events in cybersecurity and stay informed about how it might evolve. In the 1960s, a computer historian made an interesting discovery regarding the earliest cybersecurity threat. It was found that the breach of computer passwords was the primary concern at that time. In one particular incident, a researcher at the Massachusetts Institute of Technology (MIT) who had access to a mainframe system managed to gain unauthorized entry by compromising a password. MIT's time-sharing system imposed a strict four-hour time limit on the mainframe usage. To extend his allotted time, the researcher points out usernames and passwords in plain text, using them to bypass the time limit and continue using the system beyond the allocated duration [2]. The Advanced Research Projects Agency Network (ARPANET) was formed in 1967 to set the basis for cybersecurity [3]. During the 1970s, a notable advancement in cybersecurity occurred with the creation of the first computer worm. Computer worms are a form of malicious software that can self-replicate and spread across networks [4]. The first computer worm was considered to be created by Bob Thomas which infected the ARPANET computers and displayed the message I'm the creeper, catch me if you can on the screens of the affected machines [5]. In 1987, the first commercial software known as VirusScan was introduced to protect computer systems from viruses by McAfee created by John McAfee [3], and in the 1990s, the researchers at NASA developed the firewall in response to a virus attack on their base in California [5].

2.2 State of Cybersecurity

Over the decades, cybersecurity gained popularity in the personal and organizational contexts. The consequences of cyber threats such as damage of reputation, financial loss, and even the loss of lives became more apparent [1]. Cybersecurity has undergone significant changes in the last thirty years, largely due to the continuous evolution of technology and the persistent development of new attack methods by threat actors [3]. With the availability of hacking tools online for download, the barrier to entry for potential hackers has lowered, as technical knowledge is no longer a prerequisite [6]. However, as threats have evolved, so the security measures and tools.

2.3 Honeypots

A honeypot is a data framework system that acts as a decoy system to distract hackers from gaining unauthorized access to information in the system [7]. This definition incorporates two general concepts: A honeypot can be a workstation, a gadget, a server, or a whole system, and will be a distraction to an attacker and capture threat intelligence. During the action performed by the attacker, sufficient information and data are extracted and verified. According to a research paper by [6], a honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource. It can be described as a computing resource deliberately set up to be targeted and attacked. Unlike IDSs or Network Intrusion Detection Systems (NIDS), the unique value of honeypots lies in their ability to gather data on threat actors. For instance, honeypots can capture and store keystroke data from interactions between threat actors and the honeypot itself [8]. Additionally, honeypots have the potential to detect zero-day attacks, which are previously unseen and unknown exploits within the cybersecurity community. Furthermore, any attempts to access services provided by honeypots are inherently suspicious, making the data collected more likely to yield accurate and valid results [9].

2.4 Honeypot Technologies

The honeypots are categorized into high-interaction honeypots and low-interaction honeypots based on the level of interaction and the services they emulate or simulate. High-interaction honeypots replicate or emulate the full functionality of a system that almost resembles the actual production environment. While interacting with such honeypots, the attackers would get the feeling of interacting with a genuine system. However, the underlying risk associated with high-interaction honeypots is the threat actors can exploit honeypots and attack the honeypot network which is connected to [10]. It is important to consider these factors before implementing high-interaction honeypots as a means of cybersecurity strategy. On the other hand, the low-interaction honeypots emulate or simulate a very limited number of services or protocols [6], which means certain commands and functionalities will not be fully supported, emulated or simulated. Due to this limitation in their functionality, the threat actors will also have limited or fewer opportunities to interact with low-interaction honeypots. The main intention of the low-interaction honeypot is to detect suspicious activities, like login attempts or port scans directed at the honeypot system, and collect any credentials or information used by the suspicious actors [9]. Honeytrap is an example of the low-interaction honeypots that emulate services such as SSH, FTP or HTTP requests, and serve as a decoy system to attract the attackers and monitor their malicious behaviour. The honey client systems have a different purpose than traditional honeypots. The honey client, instead of passively waiting for the attackers to attack, actively searches for vulnerabilities and exploitations on their own on the system. These systems are designed as client applications and focus on identifying compromised systems or those containing malware. A specific type of Honey client is a web-based application, which aims to detect suspicious websites [11]. In addition, a honeytoken is a unique honeypot technology used to entice the threat actors with fabricated data, such as a file containing usernames or passwords. Such files are intentionally named to appear valuable to the attacker so when the attackers open the file, alerts will be triggered in the Security Information Event Management (SIEM) system of the organization. Such alerts enable security administrators to detect suspicious behaviours on the systems in a network. Tools like canarytokens from canarytokens.org can be used to create honeytokens [12].

2.5 Honeypot Software

According to the paper titled Analysis and research of a virtual honeypot framework: Honeyd[13], Honeyd is the most widely used honeypot system with low interaction and protection currently. It has mature applications in the areas of computer and network security. Honeyd does not simulate all sides of an operating system, but just the network protocol stack of an operating system for simulating specific operating systems. Honeyd works in the network layer. That intruder only can interact with the Honeyd system in the network layer. Even if Honeyd is broken, the intruders do not get access to the real system forever. However, the system can still get details of the invasion [13]. Glastpof is a low-interaction web application honeypot capable of emulating thousands of vulnerabilities to gather data from attacks that target web applications. Glastpof responds to an attacker similar to what the attacker expects from the web application [14]. Honeytrap is a low-interaction honeypot aimed to collect malware in an automated way. It provides basic functionality and imitates the behaviour of legitimate services to attract potential attackers. However, it does not fully emulate the complete functionality of real systems or services. Amun is also a low-interaction honeypot, designed to capture the malware which exploits server-based vulnerabilities; and Amun has the vulnerability module, shellcode analyzer, request handle, and Amun kernel [15]. Thug, a low-interaction honeypot, emulates a web browser to capture, analyze browser-based attacks and malware behavior. It is a valuable tool for malware analysis. However, its low-interaction nature may not provide enough data for analysis. Thug honeypot is used for client-side attacks. Thug aims to mimic the behavior of a web browser and analyze the malicious activities it encounters [16]. Cowrie is a medium-interaction SSH, and telnet honeypot used to record brute-force attacks and SSH requests. Cowrie utilizes a Python codebase, which is maintained and publicly available on GitHub. Since its source code is publicly released, not only security specialists but cybercriminals can also analyze it [17]. Kippo is a high-interaction honeypot that simulates an SSH server. Its detailed records of attacker activity specific to SSH attacks make it a valuable tool for SSH-based attack analysis. [10]. T-Pot Honeypot is a pre-configured and integrated environment for multiple honeypots. It is considered as a high-interaction; the deployment and management is suitable for organizations with limited resources. In the research conducted by [18], mentions that T-Pot is designed to be used by a distributed system of installations that forward gathered data to a community for threat intelligence. It is refined, tested and has built-in visualization to observe patterns and insights of the gathered data.

3 Methodology

3.1 Lab Environment Setup

For the experimental lab set up, the following requirements were gathered and used.

3.1.1 Tools and Systems used for Deployment

The following free and open-source software or tools were used for deployment of Honeytrap and T-Pot as shown in Table 1.

Table 1: Software used for environment setup

Software	**version**	**Remarks**
Oracle VirtualBox	6.1.24	Create virtual environment
GNS3	2.2.38	Simulate network
Ubuntu (Linux)	22.04.2 LTS	Configure Honeytrap and other servers
Debian (Linux)	11	Configure and install T-Pot
Kali Linux	2023	Attacker system
Windows	7 Professional	Client system

3.1.2 Network Topology

The virtual lab environment was created using GNS3 with the network address 172.168.30.0/24 comprising an Ethernet switch connecting with various servers such as a DHCP server, DNS server, web server, a router with an attacker, two honeypot servers i.e., T-Pot and Honeytrap. The network topology is shown in Figure 1.

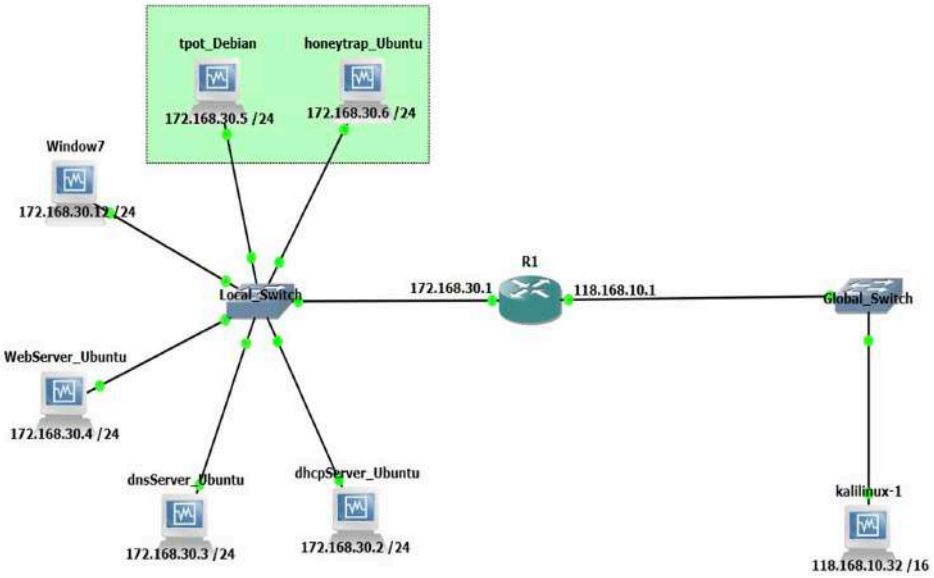


Figure 1: Network topology used for the lab set up

3.2 Installation and Configuration

3.2.1 Honeytrap

The first honeypot utilized is Honeytrap created by Tillmann Werner. According to a study on honeypot projects by [15], Honeytrap is designed to allow unauthorized individuals (referred to as blackhats) to access and operate within it, enabling the monitoring and analysis of their actions. By collecting information, the profiles and techniques of these black hats can be examined. Honeytrap is an open-source honeypot framework that is designed for capturing and analyzing network traffic to detect potential attacks. The general hardware required for running Honeytrap honeypot system is Intel Corei5 processor, 4 GB of RAM, storage of certain size to store log files, and one or more network interfaces for connectivity depending on the network topology [19]. According to the report written in [20], the installation and configuration of the Honeytrap honeypot system involves fulfilling system requirement, choosing appropriate installation methods, downloading and extracting Honeytrap from official site, installation and configuration, launching and monitoring. In some cases, advanced customization is required depending on the need of custom modules.

3.2.2 T-Pot

The second honeypot used is T-Pot, that has eighteen dockerized honeypot services. These services, along with tools such as Cockpit, Cyberchef, and ELK Stack, provide a web user interface for real-time performance monitoring, data analysis, and visualization. Moreover, T-Pot leverages Suricata, an open-source threat detection engine, to obtain information on Common Vulnerabilities and Exposures (CVE) [15][21]. As per the report from T-Pot: A Multi-Honeypot Platform, T-Pot is a comprehensive honeypot system that provides a range of services for network monitoring, attack detection, and threat intelligence gathering. The hardware requirement for running the T-Pot honeypot system is a minimum of multi-core processor with a clock speed of at least 2.0 GHz, 8 GB RAM, sufficient storage to store virtual machines and associated logs (100 GB recommended), and a reliable network interface (Gigabit Ethernet interface is typically recommended) [21]. The installation of the T-Pot honeypot system is complex compared to Honeytrap but it also varies depending on the specific environment and setup requirements. Some of the steps involved in installation and configuration of T-Pot are fulfilling the system requirements, downloading ISO image official website, setup and configuration, and finally monitoring and analysis [21].

3.3 Attack Scenario

The Honeytrap and T-Pot were installed and configured successfully with all other required operating system that serves as DHCP server, DNS server, web server, client, and the attacker. All these network components were implemented on GNS3 with all other required configured devices and cables that are used to connect servers and attackers where honeypots come in between and learn about attackers details. The network connectivity was tested by browsing webpages and pinging from server to server, and client to server. The working and functionality of deployed honeypots (Honeytrap and T-Pot) were tested through the SSH brute-force attempt, target port scanning and file access on the servers using the Kali Linux as the attacker machine.

4 Result and Discussion

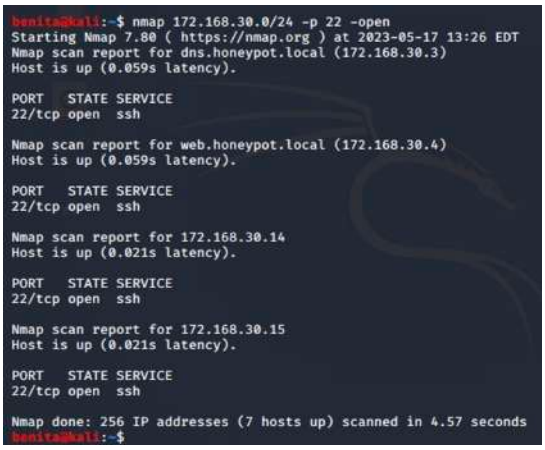
4.1 Port Scanning

The first step of a threat actor is reconnaissance, where one finds the public IP address of the victim, and then performing the scanning of the victim using tools; in this case, using an Nmap tool which is preinstalled in Kali Linux as shown in Figure 2a.

The next step would be gaining access to the victims system. Port 22 for DNS server, web server, Honeytrap, and T-pot are open. As the ports are public, anyone can have access to them. By using SSH one can see how Honeytrap and T-pot simulate operating systems. Honeytrap emulates an Ubuntu after gaining access to the honeypot. T-Pot, on the other hand, emulates Debian OS when SSH access into it is gained.

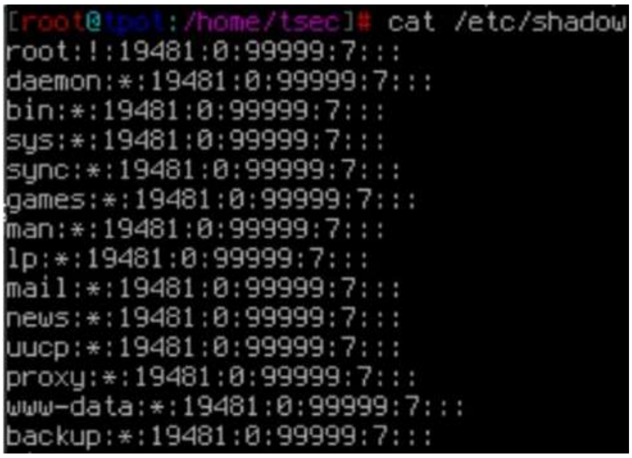
4.2 File Access

In the T-Pot it was also possible for the threat actor to access the emulated shadow file, which contains all the hashed passwords, as can be seen in the figure 2b. This gives more interaction for the threat actor with the System.



```
benita@kali:~$ nmap 172.168.30.0/24 -p 22 -open
Starting Nmap 7.80 ( https://nmap.org ) at 2023-05-17 13:26 EDT
Nmap scan report for dns.honeypot.local (172.168.30.3)
Host is up (0.059s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap scan report for web.honeypot.local (172.168.30.4)
Host is up (0.059s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap scan report for 172.168.30.14
Host is up (0.021s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap scan report for 172.168.30.15
Host is up (0.021s latency).

PORT   STATE SERVICE
22/tcp open  ssh

Nmap done: 256 IP addresses (7 hosts up) scanned in 4.57 seconds
benita@kali:~$
```

(a) Port scanning using NMAP tool



```
[root@tpot:/home/tsec]# cat /etc/shadow
root:!:19481:0:99999:7:::
daemon:*:19481:0:99999:7:::
bin:*:19481:0:99999:7:::
sys:*:19481:0:99999:7:::
sync:*:19481:0:99999:7:::
games:*:19481:0:99999:7:::
man:*:19481:0:99999:7:::
lp:*:19481:0:99999:7:::
mail:*:19481:0:99999:7:::
news:*:19481:0:99999:7:::
uucp:*:19481:0:99999:7:::
proxy:*:19481:0:99999:7:::
www-data:*:19481:0:99999:7:::
backup:*:19481:0:99999:7:::
```

(b) Content of shadow file

Figure 2: Details of port scanning and contents of shadow file

4.3 Comparative Analysis

4.3.1 Emulated Services

According to [22], Honeytrap offers a range of services for capturing and analyzing network traffic. Some of the key services provided by Honeytrap are recording and capturing network traffic, which enables the security analysts to scan the information for potential threats and intrusions. The protocols such as HTTP, FTP, SSH and SMTP are also supported by Honeytrap in addition to handling both the plaintext and encrypted traffic. It also logs captured network data and the alerting mechanisms are also offered to notify the security administrators of the potential malicious activities enabling timely response. T-Pot supports a variety of honeypots such as Cowrie (popular SSH/Telnet honeypot), Dionaea (capturing and analyzing malware samples by emulating vulnerable services such as SMB and FTP), ElasticHoney (collects data on attacks targeting web applications), Glastpof (web application honeypot that emulates vulnerable web pages to capture and analyze attacks targeting web servers), and Honeytrap (multi-purpose honeypot framework designed to capture network traffic and perform protocol-specific analysis). In addition, T-Pot supports network monitoring, threat intelligence and detailed attack reporting on the detected attacks which provides insights on techniques, tools and trends. This papers findings are in agreement with the statement confirming that Honeytrap operates as a low-interaction honeypot, simulating a limited number of services, while on the other hand, T-Pot functions as a high-interaction honeypot, offering support for a wide range of services that surpass the capabilities of Honeytrap.

4.3.2 Level of Interaction

According to [23], Honeytrap was introduced as a unique honeypot daemon with a dynamic server concept. Unlike other honeypots, Honeytrap utilizes stream monitors to examine network streams for incoming packets and activates appropriate listeners as needed. It follows a minimal engagement approach, primarily targeting casual attackers, while gathering essential data such as the attacker's IP address and attempted methods of exploitation. The goal is to collect malware in an automated fashion while minimizing any potential impact on the network's security and stability [11].

T-Pot is a high-interaction honeypot that creates a realistic environment to lure attackers and capture their actions, providing valuable insights into their techniques. It offers a wide range of services, including web servers, databases, and vulnerable applications, all designed to closely resemble genuine systems. T-Pot incorporates multiple honeypots, such as Telnet, SSH, SMB, ADB, ElasticSearch, MySQL, HTTP, SIP, MQTT, FTP, UPNP, RDP, and industrial controllers. The honeypot is extensively refined and thoroughly tested, featuring built-in visualization capabilities to view gathered data, including images. T-Pot is designed to be used as part of a distributed system, where the collected data is forwarded to a specialized threat intelligence community by default [18]. Honeytrap is indeed a low-interaction honeypot as it simulates limited services, whereas T-Pot is a high interaction that supports an enormous number of services than the Honeytrap.

5 Conclusion

Based on the study, Honeytrap tends to have a relatively straightforward installation process, with fewer dependencies and configurations required. On the other hand, T-Pot involves a more complex installation procedure due to its extensive range of supported services and the need to ensure compatibility with various components and systems. Deploying honeypots to attract and gather the data and information of the potential attackers is an effective technique for contributing to the security of networks. Upon implementing and comparing, Honeytrap as a low-interaction honeypot focuses on emulating limited services and also gathers minimal details of the potential attackers while T-Pot as a high-interaction honeypot engages attackers thoroughly and provides valuable insights into the tactics and techniques employed by the attackers. T-pot had a great capacity to record specific information and behavior patterns of attackers, allowing for more thorough investigation and comprehension of new threats, and also has a collection of different honeypots. Both honeypots have justifiable share of potential to contribute to improving and boosting network security by detecting, analyzing new attack techniques, gathering intelligence on attackers, and identifying vulnerabilities, and helping to secure the network system.

References

[1] J. Steinberg. *Cybersecurity for Dummies.* John Wiley and Sons, 2019.

[2] G. Khalil. Password security thirty-five years later. Technical report, SANS Institute. [Online]. Available: `https://www.sans.org/reading-room/whitepapers/basics/password-securitythirty-five-years-35592`.

[3] K. Chadd. The history of cybersecurity. Accessed: Feb. 28, 2021, 2020. [Online]. Available: `https://blog.avast.com/history-of-cybersecurity-avast`.

[4] K. Featherly. Arpanet. Accessed: Feb. 28, 2021, Mar. 2021. [Online]. Available: `https://www.britannica.com/topic/ARPANET`.

[5] D. Murphey. A history of information security. Accessed: Feb. 26, 2021. [Online]. Available: `https://www.ifsecglobal.com/cyber-security/a-history-of-information-security/`.

[6] L. Spitzner. Honeypots: Catching the insider threat. *IEEE Security & Privacy*, pages 99–102, 2002.

[7] M. Husak P. Sokol and F. Liptak. Deploying honeypots and honeynets. *IEEE Security & Privacy*, 2013.

[8] T. Holz and N. Provos. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection.* Addison-Wesley Professional, 2007.

[9] M. Mohssen and H. Rehnman. *Honeypots and Routers.* Auerbach Publication, New York, 2015.

[10] I. Livshitz. Whats the difference between a high interaction honeypot and a low interaction honeypot?, 2019.

[11] A. A. Lawan A. Zakari and G. Bekaroo. Towards improving the security of low-interaction honeypots: Insights from a comparative analysis. In *Proc. 1st Int. Conf. on Electrical, Electronic and Communications Engineering (ELECOM 2016)*, pages 314–321, Bagatelle, Mauritius, Nov. 25-27 2016.

[12] L. Zymberi. Honeypots: A means of sensitizing awareness of cybersecurity concerns, 2021.

[13] D. Cao L. Zhou and Y. Nian. Analysis and research of a virtual honeypot framework: Honeyd. *Computer Engineering and Applications*, 27:137–140, 2005.

[14] Kobin M. Vetsch and M. Mauer. Know your tools: Glastopf. Technical report, Honeynet Proj., 2010.

[15] Y. Aburabia and M. Omari. Network forensics tools: Honeytrap project. In *Proc. IEEE Conf. on Technologies for Homeland Security*, 2006.

[16] A. Pektas O. Erdem and A. Kara. Honeything: A new honeypot design for cpe devices. *KSII Trans. on Internet and Information Systems*, 12(9), 2018.

[17] Warren Z Cabral, Craig Valli, Leslie F Sikos, and Samuel G Wakeling. Advanced cowrie configuration to increase honeypot deceptiveness. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 317–331. Springer, 2021.

[18] Vesselin Bontchev and Veneta Yosifova. Analysis of the global attack landscape using data from a telnet honeypot. *Information & Security: An International Journal*, 43(2):264–282, 2019.

[19] P. Martini F. Leder and M. Spreitzenbarth. An overview of honeypots and their use in security research and development. *Int. J. of Advances in Security*, 4(3), 2011.

[20] Y. Wang Z. Tang Y. Li, P. Zhang and Y. Xue. An intelligent honeypot system for detecting and analyzing modern web attacks. *IEEE Access*, 8, 2020.

[21] S. Wallner D. Wegemer, L. Rist and E. R. Weippl. T-pot: A multi-honeypot platform. In *Proc. 13th Int. Conf. on Availability, Reliability, and Security (ARES)*, 2018.

[22] M. Zulkernine and Bhagyanvai. Honeypots: Concepts, approaches, and challenges, 2014.

[23] T. C. Schmidt C. Keil M. Nawrocki, M. Wählisch and J. Schönfelder. A survey on honeypot software and data analysis. *arXiv preprint arXiv:1608.06249*, 2016.