Date of publication DEC-31, 2024, date of current version APR-09, 2024.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.23.4.3761

Implementing Honeypots for Detecting Cyber Threats with AWS using the ELK

VIKTOR KOSHELIUK, YURII TULASHVILI
Lutsk National Technical University, Volyn, 43026, Ukraine

Corresponding author: Viktor Kosheliuk (e-mail: viktor.koshelyuk@gmail.com).

ABSTRACT The growing need to use cloud computing to design information systems that are accessible 24/7 opens up a great opportunity for potential attacks by malicious actors. Every day, we see a large number of cyberattacks in all aspects of life. One of the methods of solving the problem of countering hackers is to protect the server using a honeypot. The proliferation of multi-level honeypots characterizes one of the methods of detecting and preventing the actions of criminals by generating a fake server to redirect hacker attacks. In our work, we propose to use honeypots as an element of IT infrastructure intelligence to identify vulnerabilities and study patterns of potential attacks. To achieve this goal, we deployed honeypots in five different regions of the AWS cloud provider. The data obtained was analyzed using ELK Stack (elasticsearch, logstash, kibana). The integration of honeypot and ELK Stack demonstrates an effective solution for detecting potential attacks by providing a detailed visualization of the behavior of attackers.

KEYWORDS T-Pot; cloud computing; cybersecurity; ELK stack; honeypot.

I. INTRODUCTION

IN today's world of global interaction, the use of cloud computing infrastructure is growing exponentially. Cloud infrastructure technologies determine the strategy of access to the information system, which provides a fast network connection as needed and required by users. A generated request for an information product can be easily obtained and used with minimal intervention from administrative staff or involvement of a service provider. Many of us will witness fundamental changes in information technology throughout our lifetime. Ongoing changes and developments in information technology are having a significant impact on how this happens, as well as on the concepts of capital and value added of information systems. Cloud infrastructure services are often located in different premises or network segments and are remotely accessible 24/7 to cloud users.

The technology of using cloud services is a necessary component of business management in many organizations. It creates a competitive advantage in implementing efficient operations. This is an evolution of the use and demand for information technology in organizations that are turning cloud services into a dominant business model for technology and innovation resources. Despite the tremendous benefits and opportunities, the use of cloud technologies makes it inherently valuable to understand the pitfalls and drawbacks associated with widespread cloud services, such as customer data protection and security. The transformations associated with the use of cloud services offer tremendous opportunities to transform business management practices in companies, creating a competitive advantage that has led many companies to adopt this technology as a solution to manage their business.

The fundamental characteristics of cloud computing services include on-demand self-service, broad network access, resource pooling, measurable service, rapid elasticity and location independence. Despite these benefits, the widespread adoption of this new technology faces several obstacles, including security and privacy concerns in addition to traditional security risks. Many companies have documented various studies in reports. They reflect the rise of security incidents that go undetected by existing security mechanisms, and the number of successful thefts and frauds is on the rise.

According to GDATA [1], there is an increase in new threats and attacks on existing cloud services. Studies have also revealed a large number of malicious software. In the case of large amounts of information or new types of attacks, traditional security tools have limited capabilities to ensure the security of the information system. The application of the expert approach takes a lot of time, and attacks are difficult to identify, which are its main disadvantages [2]. In addition to detecting and stopping attacks on information systems, understanding the motivation, goals and strategies of attackers is important for finding new ones and predicting attacks that may be carried out against cloud infrastructure systems.

To protect against cyberattacks and configure a secure IT infrastructure, there are various traditional approaches, including in-depth perimeter security settings deployed by cloud service providers. IT security instantly becomes an issue for anyone who connects his or her system to the global network. Threats to the safe functioning of infrastructure range from hacker intrusions, denial of service attacks to computer worms, viruses, etc. We have to understand that an intrusion into a network or system can never be eliminated, but it can be mitigated. The technologies and methods of computer criminals are constantly improving.

Given the complex nature of cyberattacks and their growing tendency to develop in stages, it is necessary to think outside the box and explore unconventional approaches to preventing, detecting and recovering IT infrastructure from cyberattacks. One approach is to integrate active defence mechanisms into the cybersecurity infrastructure, which allows attacks to be monitored in a controlled environment to research and learn attack patterns. The introduction of active protection mechanisms into the existing infrastructure makes it possible to prepare appropriate measures to counter possible incidents. To eliminate the shortcomings of traditional threat detection and prevention systems, since 1992, it has been proposed to use honeypot [3] as a powerful data system that monitors, detects and analyses malicious behavior. Honeypot systems are designed to minimize detection, attack, or compromise [4]. By using a decoy, a network administrator can determine the identity, intentions and strategies used by attackers to penetrate the system, as well as the types of attacks that were used.

Honeypot collects a small amount of information, which is analyzed to build statistics on the methods used by attackers and to determine whether there are any new solutions that can be used to combat them. Analyzing IT infrastructure cyberattack patterns can be useful for configuring a network asset protection system [5]. Honeypots collect as much information about the attack as possible. The honeypot should operate in stealth mode so that the attacker is unaware of its presence.

The value of a honeypot as a security resource is that it can be verified, attacked, or hacked. This means that regardless of what we call a decoy, we hope and aim to expose vulnerabilities by attacking the system [6]. Honeypot describes a tool for detecting and responding to potential cybercrime. Since a honeypot cannot prevent a specific intrusion or virus spread, it only collects information and identifies patterns of attacks and possible incidents. Implementing decoys opens up ways to increase the level of protection and counter future security threats. The concept behind decoys is to allow the hacker community to spend time and resources attacking decoys rather than the organization’s IT infrastructure [7]. The attacker is identified and misled to attack the decoy, thus protecting the organization’s infrastructures from an attack.

Based on the information gathered by the decoys, an organization’s IT department will have a better understanding of hackers' attack patterns, motivations, and how they operate. With all this knowledge of potential threats, an organization can better prepare to arm itself with the necessary defense and processes. From decoy information, organizations can better understand three important security concepts: prevention, detection and response.

II. METHODS OF RESEARCH

The concept of cloud computing proposes a computing paradigm where physical resources can be made available to different users on the one hand, and customers have their own computing spaces on the other hand, using virtualization techniques. The general concept of cloud computing is that services and resources are provided by a CSP over broadband networks, mainly the Internet, and customers use the resources and services as they need them and pay only for what they consume.

Cloud computing can be classified [8] based on two types of models: service model and deployment model. Depending on several factors, such as business requirements, organizational capability specifications, storage requirements, costs, etc., enterprises choose different combinations of these models.

There are three types of cloud computing service models offered, which are critical in determining how they are used and influence an organization. Various factors, practices, and requirements influence the type of cloud computing service an organization uses [9]. Cloud computing services have no limitations compared to the on-premises model, where organizations must manage and maintain every component of the IT system, including applications, data, servers, storage, virtualization, networking, and middleware. As a solution to meet the demand for IT resources, cloud service models include infrastructure as a service (IaaS); platform as a service (PaaS); software as a service (SaaS).

The most common classification involves the intensity of the attacker's interaction with the existing infrastructure. The amount of data at different levels of an organization’s functioning differs significantly. A higher level of infrastructure operation poses a greater risk to the security of the company's network. Depending on the amount of data and the level of interaction between the attacker and the system, there are low-interaction, intermediate-interaction, and high-interaction honeypots [10].

A low-interaction honeypot (LIH) uses one or more simple services that record all communication attempts to specific services, such as a web server or SSH server. They only mimic a set of operating system services and resources. This makes honeypots easy to deploy and maintain. These types of decoys are typically used to passively collect statistical data on network traffic changes and identify the attacker. Examples of such low-interaction tools include Honeytrap, Spectre, and KFsensor [11].

Medium-Interaction honeypot (MIH). Mid-level decoys are more complex than low-interaction decoys. At this level, a better illusion of an operating system is created, as the attacker can interact with services and protocols in the middle layers of the OSI model. The medium level of involvement is enough to interest attackers and gives the administrator the right to choose which services to simulate to better understand how to use them to detect the attacker. An interested attacker discovers more apparently open operating system services and the likelihood that the attacker will find a vulnerability increases, but it is still unlikely that the system will be compromised. Increased interactivity also allows more sophisticated attacks to be recorded and analyzed. Medium-interaction decoys are used to detect and process botnets and collect malware. Examples of such medium-interaction tools include Cowrie and HoneyPy.

Highly interactive lures mimic the functioning of a computer system that uses all operating system services. This type of decoy is more advanced than low- and medium-interaction decoys because it provides an imitation of the operating system. The purpose of using this type of bait is to provide the attacker with a real system to interact with without restrictions or simulation to obtain a comparison of large amounts of attacker data, as all actions can be recorded and analyzed. Using a high-interaction decoy makes it possible to force attackers to interact with fake operating system data, applications and/or services over a longer period in order to collect and analyze a wide range of attacker information and intelligence, such as attacker intent, behavior, malware, commands, keystrokes and software tools used. Since the attacker has a large amount of resources at his disposal, the risk associated with this level of honeypot is very high and should always be monitored to ensure that it does not become a threat and should be used for research purposes and not used in production. An example of a high interaction tool is honeynet.

A comprehensive approach to analyzing and consolidating honeypot intercepted traffic data is offered by the ELK (Elasticsearch, Logstash and Kibana) open-source platform, where each component performs its own role. The ELK stack architecture and file processing algorithm involves the interaction of ELK Logstash components from different sources and the connection through an intermediary. The next step is the processing of unstructured data by the Logstash configuration for parsing. From Logstash, the processed data is indexed in Elasticsearch. At any time, the files stored in Elasticsearch can be visualized in a browser using Kibana on the main graphical user interface.

Logstash is used as a log management tool for centralized logging and analysis of logs. The general purpose of Logstash is to collect unstructured data from source streams of information, parsing it according to a set of filtering rules. At the same time, the Logstash package is used to output the processed data for additional analysis and storage. To perform data processing, the configuration file *.conf is set up. Several configuration files can be created simultaneously for different input data.

The Logstash configuration file contains three sections: settings for incoming data streams (input); a parsing filter configuration script for processing structured information from unstructured data (filter); settings for outputting data that has been processed using filters (output). To change the log processing, standard Logstash plugins are used in the configuration file. The contents of the Logstash configuration file are used to filter the data.

One of the most effective solutions for visualizing all ELK stack events is the T-Pot system, which contains a wide range of honeypots and security tools that can be deployed on a virtual machine. One of the most effective solutions for visualizing all ELK stack events is the T-Pot system, which contains a wide range of honeypots and security tools that can be deployed on a virtual machine. Developed by Deutsche Telekom's security team, T-Pot is a distributed honeypot platform that provides the implementation of more than 20 honeypots and a significant number of visualization options using ELK, real-time animated attack maps and many security tools for downstream services and operating system services. The basic idea behind T-Pot is to create a system that defines the entire TCP network range as well as individual UDP services as a honeypot. The decoy framework contains implementations for various protocols and services (e.g., SSH and telnet (cowrie), HTTP and FTP (dionaea), SMTP (mailoney), Intrusion Detection System (suricata), etc).

The first type of honeypot was developed in 1997 and was called the Deceptive Toolkit, with the main idea being to use deception to attack back. In 1998, the first commercial honeypot appeared, called Cybercop Sting, and since 2002 it has been freely distributed and used. Since then, honeypot technology has improved significantly and many experts believe that this is only the beginning. In 2005, the Philippines launched a government initiative to use Honeypot, which was launched to promote computer security in the Philippines.

Researchers and IT infrastructure security experts believe that honeypots are decoy resources. The main purpose of using a decoy is to simulate the functioning of a real system in order to gather information about threats. For an attacker, a honeypot simulates the functioning of a particular service. Decoys help to strengthen defenses against cyberattacks and are often used as an early warning system for zero-day exploits or to collect malware samples from botnets. For example, they are deployed as an additional layer of penetration detection, to support efforts against distributed denial of service (DDoS) attacks, to identify attackers, or to analyze the operation of new malware [12].

The work of researchers [13] presented an experimental cybersecurity architecture using the Docker container management platform. The software tools for implementing the architecture have the properties of scalability and flexibility, which is necessary for cyber simulation. The function of dynamically changing the parameters of the created decoy allows configuring the topology of cyber nodes, the software environment, and supports the configuration of the main indicators of the experiment [14]. Configuration flexibility allows for real-time changes to the display, thereby reducing the overall cost and facilitating the analysis process.

One of the decoy deployment methods was investigated in Vestergaard [15]. The proposed deployment technology uses container-based environments, as its GitHub repository contains a dockerfile to create a corresponding Docker image. The toolkit for creating the base image was implemented using python:3.7-slim-stretch with the resulting Docker image having only 145 MB.

The research decoy in [16] uses the T-Pot software over a two-week period in Europe, powered by Google Cloud Platform. The results of the decoy showed that Cowrie was the most attacked honeypot with 102,000 connections over a 14-day period (35% originating from China). At the same time, it is worth noting that the Dionaea lure was not attacked.

Paper [17] presented a mechanism for creating an enterprise decoy to protect a virtual machine in a cloud infrastructure. By using a Honeyed honeypot with Snort, hidden security vulnerabilities can be detected and prevented from being intruded upon or exploited by attackers. According to the results of the experiment, the corporate honeypot was effectively implemented into the security mechanisms of the organization’s infrastructure. The information collected by snort is useful for studying and analyzing the behavior of the intruder.

Sophos [18] carried out experimental studies on the AWS platform of deploying SSH bait. The main purpose of the experiment was to establish the dependence of the bait on the hosting provider. Sophos found that 95.4 per cent of the traffic captured in their decoys originated from China. The average number of attacks is approximately 757 per hour. The Sophos researchers also reported a huge number of SSH login attempts using default credentials, with the majority using root and password 123456. The time of the first attack was also measured, with the shortest time being against the São Paulo server (52 seconds) and the longest against the Ireland server (1 hour and 45 minutes).

Chen et al. [19] showed that deployment technology using a container management platform creates problems with collecting Docker cluster records. The paper proposed to use ELK, Filebeat Kafka to develop a system collector and parser for Docker containers in order to generate ELK-supported records in real time. Filtering and sending data visualizations for analysis has significantly increased staff efficiency. The result is high real-time performance, stability, and high availability of the decoy operation.

III. METHODOLOGY

The experimental study is based on the use of IaaS and containers. In our work, we used a combination of the most popular low- and medium-interaction decoys mentioned in [20]. High-interaction decoys are characterized by a high approximation to the real production environment, but also require the use of more resources and increase security requirements for the testing environment [21]. Taking into account the efficiency of using operating costs during the study, images of the following docker containers that are part of the T-pot were used to monitor and analyze anomalous traffic: ciscoasa [22], cowrie [23], dionaea [24], glutton [25], honeytrap [26].

One of the objectives of the study is to investigate the difference in attacker activity in different geographical regions of IaaS. Cloud service providers provide the ability to choose the geographic location of the host and the IP address range. Amazon Web Service (AWS) was chosen to deploy the decoys in different regions. Among the AWS services for the host, Amazon EKS was chosen to manage the docker-catenators. Amazon EKS manages a single Kubernetes control panel for each cluster, and the control plane infrastructure is not shared between clusters or AWS accounts. Amazon EKS runs up-to-date versions of the open-source Kubernetes software, so we can use all existing plugins and tools from the Kubernetes community. This means that we can easily move any standard Kubernetes application to Amazon EKS without any code changes. Fig. 1 shows a diagram of how the cloud provider's services interact to deploy decoys on each host [27].

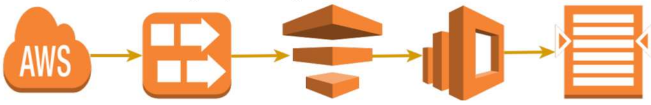

Figure 1. Model of interaction and deployment of AWS services with ELK Stack

AWS CodePipeline automatically builds, tests, and runs an application every time the code changes; the developer uses a graphical user interface to model workflow configurations for the release process in the pipeline. AWS CodePipeline integrates with several Amazon services. It gets source code from Amazon Simple Storage Service and deploys both AWS CodeDeploy and AWS Elastic Beanstalk. The developer can also integrate AWS Lambda functions or third-party DevOps tools such as GitHub or Jenkins. AWS CodePipeline also supports custom systems and actions through the AWS command line interface. These custom actions include build, deploy, test, and invoke, which facilitates unique release processes.

For the experiment, we created an EC2 instance in each region with t3.xlarge parameters, which follows the T-Pot deployment guidelines. Each instance had Debian 12 (HVM) as the operating system and was equipped with 8 GB of RAM, 2 vCPUs, and up to 5 Gigabits of network bandwidth. The experiment design involved deploying instances in different regions to study intruder detection. We deployed the distributed architecture as shown in Fig. 2 in the regions of the cloud service provider's operation: US East (N. Virginia), Asia Pacific (Tokyo), Canada (Central), Europe (Frankfurt), South America (São Paulo). According to the AWS classification, the instance distribution was carried out in accordance with the geographical location of us-east-1, ap-northeast-1, ca-central-1, eu-central-1, sa-east-1.

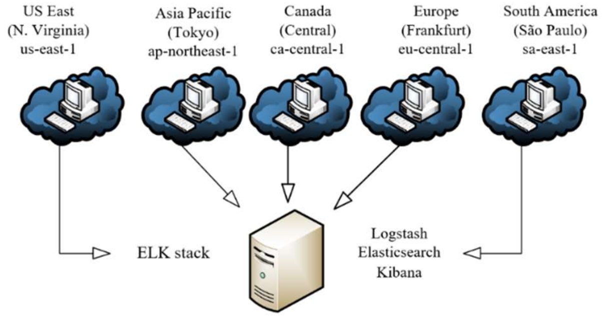

Figure 2. Network diagram of the experiment

In the distributed architecture, the management console is located locally to prevent a possible attack on the internal network of the experiment. The configuration of firewall rules provides for no priority of access to the ELK stack from other hosts in the local segment. Monitoring and analysis of traffic using honeypots involves the use of the most popular SSH (on port 22) and HTTP/HTTPS (on ports 80 and 443) services. To monitor the presence of an intruder, each honeypot runs several services to monitor and collect network traffic for the corresponding port. Table 1 shows the open ports for the different honeypots.

Table 1. Open ports in the honeypot architecture

Honepots	Open ports
ciscoasa	22, 443, 500, 514, 4500
cowrie	22, 23
dionaea	21, 23, 80, 135, 445, 1433, 3306, 5060, 5900
glutton	80, 443
honeytrap	21, 22, 23, 25, 80, 135, 443, 445, 5060, 5900

To protect the ELK stack, ports 22, 80, and 443 should not be used to access the main dashboard. T-Pot configuration recommendations include using port 64297 via HTTPS for secure access to the dashboard and port 64295 for SSH access. Inbound traffic configuration has been allowed for the port range 0-64000 in order to receive data to various decoys. Ports above 6400 were reserved for managing the experimental infrastructure.

Logging of the data obtained from honeypots was carried out using Logstash. To analyze and visualize the information we received, we used Elasticsearch and Kibana, which is part of the Elastic Stack. At the next stage, the data from Logstash and Elasticsearch is transferred to Kibana. The ELK stack management console connects to remote instances by deploying honeypots to collect data on possible attacks.

IV. RESULTS AND DISCUSSION

This section presents the results of monitoring and analysis of data collected in different geographical regions during the weekly period from 14.01.2024 to 21.01.2024. The analysis of logs generated during this period showed the following results (see Fig. 3): 758 attacks in ciscoasa lures, 207,864 attacks in cowrie lures, 315,148 attacks in dionaea lures, 37563 attacks in glutton lures, 7514 attacks in honeytrap lures. It should be noted that the largest number of attacks was detected with cowrie, dionaea, and glutton baits. This confirms the high vulnerability of ports of popular operating system services that use SSH and Telnet protocols. The experiment confirms the statement that even emulation and a combination of replacing the port number of the respective services is not an obstacle to an attack.

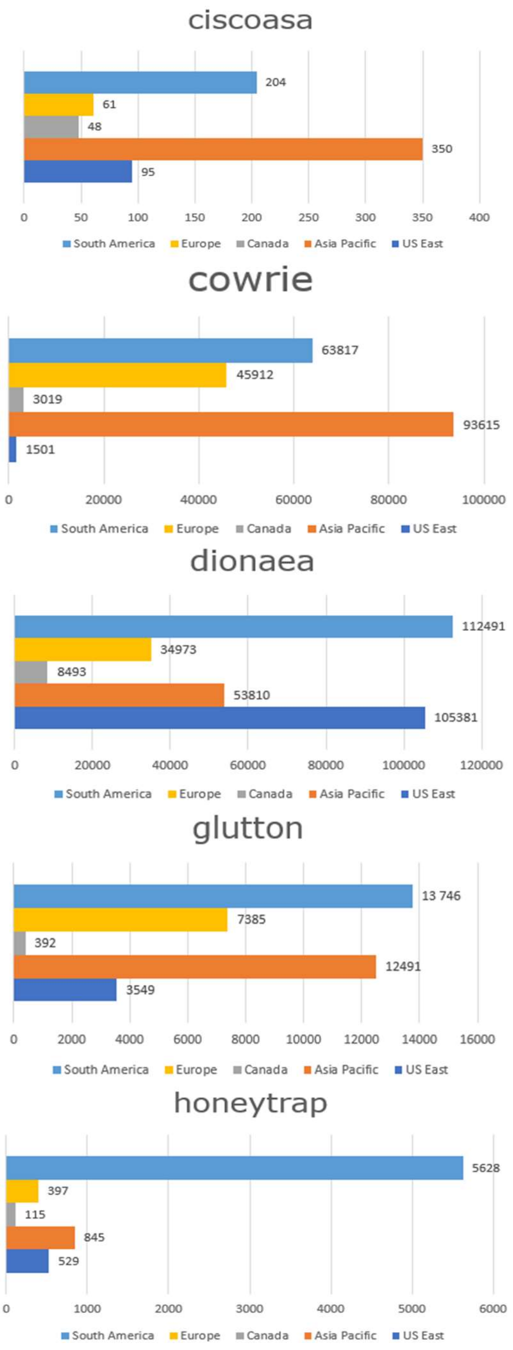

Figure 3. Analysis of the activity of honeypots by type

A breakdown of attack detections by geographic region showed the following results: US East detected 111055 attacks, Asia Pacific detected 161 111 attacks, Canada detected 12067 attacks, Europe detected 88728 attacks, South America detected 195 886 attacks. The analysis shows that the majority of attacks come from Asia and South America. However, it is important to take into account the limitations of the data obtained, which only records the apparent origin of the attack. With the use of VPN technology or the use of compromised systems as intermediaries, the data may provide an inaccurate picture of the source of attacks. These circumstances highlight the need for further research and analysis to draw more accurate conclusions about the origin of attacks. Fig. 4 presents a graphical representation of the results of the information and data collection process.

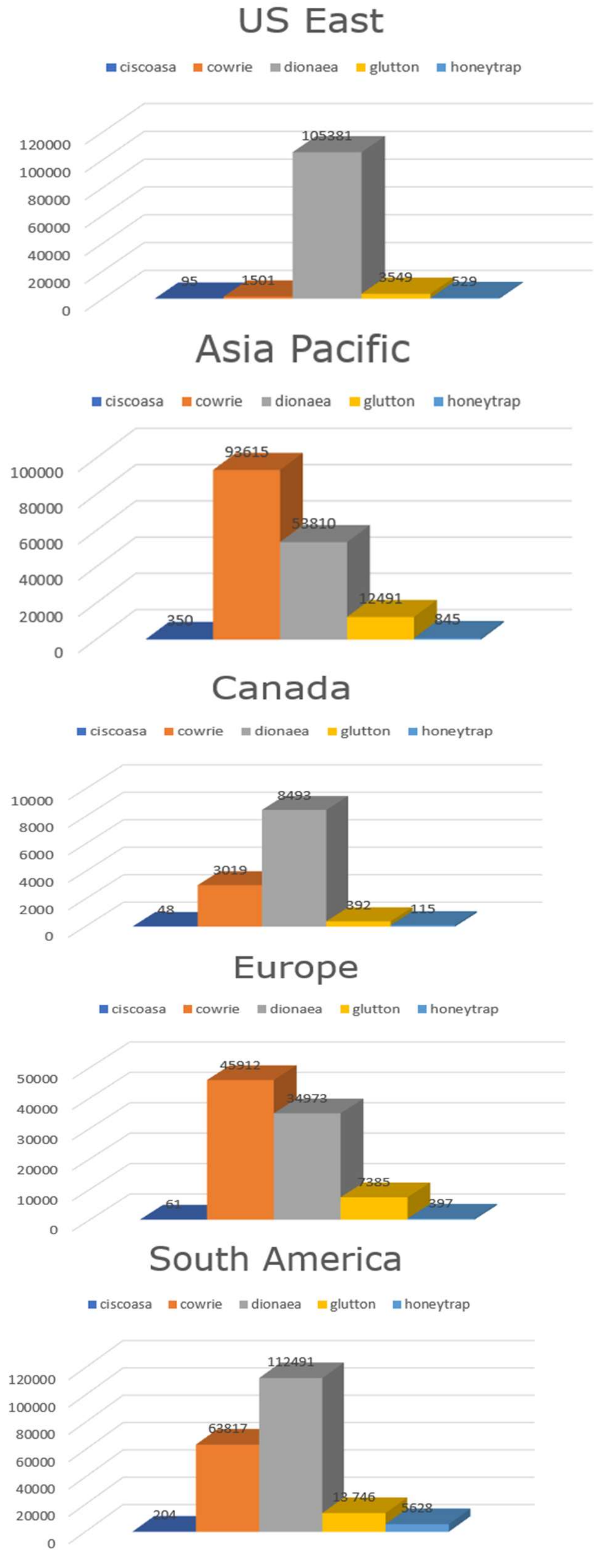

Figure 4. Comparison of honeypots by region of distribution

Honeypots cannot predict an unexpected attack, but they can help detect it. In some cases, this can prevent attackers from attacking the server directly. Detecting potential attacks helps to protect our infrastructure by applying appropriate firewall rules, creating strong passwords, and configuring encryption, digital signatures, and authentication technologies. Honeypots provide intruder detection, similar to the functionality of intrusion detection systems to protect facilities in the event of criminal activity.

V. CONCLUSION AND FUTURE WORK

The ability to adapt to any changes by automatically deploying and using resources on demand makes cloud computing an important element of IT infrastructure for all organizations, including SMEs. The growing popularity of cloud computing carries significant risks, as it attracts the attention of attackers.

In our work, we used the T-pot platform to create a threat analysis model. This model highlights the importance of understanding attacks, behaviors, and patterns for all organizations. For us, it is critical to understand the behavior of attackers in the context of cyberattacks. Hence, we define an attack model based on attacks and behaviors. However, this model only works effectively if there is a significant amount of data related to network incidents. To analyze cyber threats, we used honeypot data collected using AWS. This data was analyzed using the ELK stack to visualize log data. It is important to note that ELK uses elasticsearch to identify different types of cyber incidents. In order to study the methods used by attackers, the most typical honeypots were deployed in different regions: Europe, Asia, North and South America. Our analysis shows regional differences in the activity of threat actors during data collection.

It has become apparent that attackers are constantly targeting honeypots. Most attacks on cloud infrastructure are similar in type, as attackers attempt to gain access to services and services across the system. Experimenting with low- to medium-interaction decoys is valuable because it can be used to detect and mitigate future cyberattacks. The main advantage of using such decoys for threat analysis is that there are no harmful effects on the functioning of the main system. Such analysis can be effective in the design and development of IDS and IPS.

Honeypot is generally considered to be a flexible cyberattack prevention technology that is useful and functional for a large number of different situations. Honeypots are used in different forms and in different cases to detect intruders. If a simple, low-interaction honeypot is required to avoid the high risks associated with a high-interaction honeypot, the purpose and need for using such a honeypot is determined by the specific circumstances and security objectives. There are many benefits to using decoys in cloud infrastructure. In particular, they analyze data with a high level of trust. In addition, they are considered to be uncomplicated devices that can operate effectively in resource-intensive environments. At the same time, they help in monitoring and detecting unauthorized activities.

In future work, we intend to extend the cyberattack model. One aspect of this extension could be to set up decoys to obtain attack data and use different ISPs in typical geographic regions. This attack data can be analyzed using appropriate tools to identify attack patterns. The resulting patterns can be used to train IDS and IPS systems to automate future processes. These attack patterns can be used to implement cyber threat hunting techniques to better understand cyberattacks in a given geographic region.

References

[1] P. S. Negi, A. Garg and R. Lal, “Intrusion detection and prevention using honeypot network for cloud security,” *Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2020, pp. 129-132, https://doi.org/10.1109/Confluence47617.2020.9057961.

[2] J. Chacon, S. McKeown and R. Macfarlane, “Towards identifying human actions, intent, and severity of APT attacks applying deception techniques – An experiment,” *Proceedings of the 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, Dublin, Ireland, 2020, pp. 1-8, https://doi.org/10.1109/CyberSecurity49315.2020.9138859.

[3] S. Ravji and M. Ali, “Integrated intrusion detection and prevention system with honeypot in cloud computing,” *Proceedings of the 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, Southend, UK, 2018, pp. 95-100, https://doi.org/10.1109/iCCECOME.2018.8658593.

[4] S. Govindaraj, S. Prakash, *Joint Honeypot Networks and Hybrid Intrusion Detection System for Mobile Cloud Computing*, Master thesis, Dublin, National College of Ireland, 2020. https://norma.ncirl.ie/id/eprint/4171.

[5] S. Lysenko, K. Bobrovnikova, V. Kharchenko, O. Savenko, “IoT multi-vector cyberattack detection based on machine learning algorithms: traffic features analysis, experiments, and efficiency,” *Algorithms*, vol. 15, 239, 2022. https://doi.org/10.3390/a15070239.

[6] R. Guan, L. Li, T. Wang, Y. Qin, W. Xiong and Q. Liu, “A Bayesian improved defense model for deceptive attack in honeypot-enabled networks,” *Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Zhangjiajie, China, 2019, pp. 208-214, ttps://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00043.

[7] M. Dawood, S. Tu, C. Xiao, H. Alasmary, M. Waqas, S. Ur Rehman, “Cyberattacks and security of cloud computing: A complete guideline,” *Symmetry*, vol. 15, no. 11, 1981, 2023. https://doi.org/10.3390/sym15111981.

[8] C. Kelly, N. Pitropakis, A. Mylonas, S. Mckeown, W. Buchanan, “A comparative analysis of honeypots on different cloud platforms,” *Sensors*, vol. 21, 2021. https://doi.org/10.3390/s21072433.

[9] S. B. Goyal, P. Bedi, S. Kumar, J. Kumar, N. R. Karahroudi, “Application of deep learning in honeypot network for cloud intrusion detection,” In: Chaki, N., Devarakonda, N., Cortesi, A., Seetha, H. (eds), *Proceedings of International Conference on Computational Intelligence and Data Engineering. Lecture Notes on Data Engineering and Communications Technologies*, vol 99, 2022. Springer, Singapore. https://doi.org/10.1007/978-981-16-7182-1_21.

[10] C. Gupta, T. Van Ede and A. Continella, “HoneyKube: Designing and deploying a microservices-based web honeypot,” *Proceedings of the 2023 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, 2023, pp. 1-11, https://doi.org/10.1109/SPW59333.2023.00005.

[11] N. El Kamel, et al., “A smart agent design for cyber security based on honeypot and machine learning,” *Security and Communication Networks*, 2020, pp. 1-9. https://doi.org/10.1155/2020/8865474.

[12] A. Vetterl, Honeypots in the Age of Universal Attacks and the Internet of Things, Technical Report UCAM-CL-TR-944, University of Cambridge, Computer Laboratory, February 2020. https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-944.pdf.

[13] K. Bobrovnikova, S. Lysenko, B. Savenko, P. Gaj, O. Savenko, “Technique for IoT malware detection based on control flow graph analysis,” Radioelectronic and Computer Systems, no. 1, pp. 141-153, 2022. https://doi.org/10.32620/reks.2022.1.11.

[14] G. Márquez, H. Astudillo, “Identifying availability tactics to support security architectural designs in microservices-based systems,” Proceedings of the 13th European Conference on Software Architecture ECSA’19, Association for Computing Machinery, New York, NY, USA, 2019, vol. 2, pp. 123-129, https://doi.org/10.1145/3344948.3344996.

[15] M. R. Amal, P. Venkadesh, “H-DOCTOR: Honeypot based firewall tuning for attack prevention,” *Measurement: Sensors*, vol. 25, 100664, 2023, https://doi.org/10.1016/j.measen.2022.100664.

[16] D. Le, A. Zincir-Heywood, “Exploring anomalous behaviour detection and classification for insider threat identification: Anomaly detection and classification for insider threat identification,” *International Journal of*

Network Management, vol. 31, e2109, 2020. https://doi.org/10.1002/nem.2109.

[17] I. Livshitz, *Low, Medium and High Interaction Honeypot Security*, 2019, [Online]. Available at: https://www.akamai.com/blog/security/high-interaction-honeypot-versus-low-interaction-honeypot-comparison.

[18] M. Boddy, *Exposed: Cyberattacks on Cloud Honeypots*, 2020, [Online]. Available at: https://www.sophos.com/en-us/medialibrary/PDFs/Whitepaper/sophos-exposed-cyberattacks-on-cloud-honeypots-wp.pdf.

[19] C.-A. Chen, "With great abstraction comes great responsibility: Sealing the microservices attack surface," *Proceedings of the 2019 IEEE Cybersecurity Development (SecDev)*, Tysons Corner, VA, USA, 2019, pp. 144-144, https://doi.org/10.1109/SecDev.2019.00027.

[20] R. Kumar, R. Goyal, "Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC)," *Computers & Security*, vol. 97, 101967, 2020. https://doi.org/10.1016/j.cose.2020.101967.

[21] U. J. C. Pramodya, et al., "Agenthunt: Honeypot and IDS based network monitoring device to secure home networks," In: *Proceedings of the Future Technologies Conference (FTC) 2021, Volume 3*, Springer International Publishing, 2022, pp. 194-207. https://doi.org/10.1007/978-3-030-89912-7_16.

[22] Cymmetria Research. Cisco ASA honeypot. 2024, [Online]. Available at: https://github.com/Cymmetria/ciscoasa_honeypot.

[23] Michel Oosterhof. Cowrie SSH/Telnet Honeypot. 2024, [Online]. Available at: https://github.com/cowrie/cowrie.

[24] Dionaea. dionaea - catches bugs. 2024, [Online]. Available at: https://github.com/DinoTools/dionaea.

[25] L. Rist, J. Vestergaard, D. Haslinger, A. Pasquale, and J. Smith, Glutton: low-interaction honeypot, 2024, [Online]. Available at: https://github.com/mushorg/glutton.

[26] T. Werner, Honeytrap, 2024, [Online]. Available at: https://github.com/armedpot/honeytrap.

[27] Elasticsearch, The Elastic Stack, 2024, [Online]. Available at: https://www.elastic.co/elastic-stack.

***VIKTOR KOSHELIUK,** an Assistant Professor, PhD, Department of Computer Science, Lutsk National Technical University, Lutsk, Ukraine. Field of scientific interests: information systems design, analytics of information security, cloud technologies, cyber security of information technologies, incident handling and response.*

https://orcid.org/0000-0002-4136-5087
email: viktor.koshelyuk@gmail.com

***PROF. YURII TULASHVILI,** Doctor of Pedagogical Sciences, a Professor, Department of Computer Science, Lutsk National Technical University, Lutsk, Ukraine. Field of scientific interests: mathematical modeling of complex systems, software development and analysis algorithms, study of the effectiveness of the use of computer information technologies in education, science and production.*

https://orcid.org/0000-0002-0780-9529
email: y.tulashvili@ukr.net