Implementation of Cowrie Honeypot System and Improvement of Log Analysis

Milan Panić[1], Nemanja Maček[2]

[1] *Pan-European University "APEIRON", Faculty of Information Technology, Banja Luka, Bosnia and Herzegovina, milan.v.panic@apeiron-edu.eu, ORCID ID: 0009-0006-9270-7354*

[2] *University Business Academy in Novi Sad, Faculty of Social Sciences, Belgrade, Republic of Serbia, maceknemanja@gmail.com, ORCID ID: 0000-0002-3465-7524*

Professional paper

https://doi.org/10.7251/JIT2601069P UDC: 37.016::003-028.3

Abstract: This paper aims to explain how honeypots work, how they are implemented, and why they have become a key aspect of cybersecurity. Honeypots are capable of doing everything from detecting new attacks never seen before in their environment to tracking programmed credit card fraud and identity theft. The paper implements the Cowrie honeypot system in a controlled environment to simulate attacks on SSH and Telnet services. Special focus is placed on the analysis of generated JSON log records, the complex structure of which makes forensic processing difficult. As a contribution to the paper, a Python helper module has been developed to convert raw log files into a readable and structured text format, thus improving the efficiency of security event analysis.

Keywords: Cowrie, honeypot, SSH, Telnet, log

Introduction

High-tech crime represents one of the biggest security challenges in the world. The development of information technologies, in addition to the benefits that it has improved the operation of many systems, has also led to the development of new forms of crime. The previous decade has been marked by a constant fight against new forms of high-tech crime, an increase in crimes in this area, but also by the constant strengthening of the capacities of departments for combating high-tech crime.

In order to create a system that is interesting enough for hackers to carry out an attack, we must create a sufficiently tempting system. They will try to gain access by using security vulnerabilities in the system. By following hackers, we are not sure whether we will be the ones who will have control. It is necessary to determine the following:

- Does the hacker know that this is a real system or is it a honeypot?
- Is he aware of how great a tool it is for administrators to obtain information about security vulnerabilities in the system, but also about the attackers themselves?
- What is gained from hacking such a system?

A (high-level) hacker is often like a cat, he enters quietly, leaves small traces and makes the whole game an art of evasion. A digital forensics expert in this context is like a pathology technician, when a hacker makes a mess, the forensics expert comes in and looks at the situation from the beginning, reads the log files like fingerprints. Accordingly, it is necessary to mention the cybersecurity administrator who is not just a "gatekeeper", he already knows where the keys are hidden and how to quickly take them away from the attacker. When such parties meet, then it is no longer just a clash of knowledge, but a dance of roles. In some cases, the administrator becomes a hunter who does not shoot, but sets traps (honeypots). From this aspect, digital forensics plays the role of an archivist and judge, i.e. records either defeat or victory, while the administrator acts as a battlefield that the hacker must bypass. It is an eternal struggle in which every move quickly changes the rules of the game.

METHODS AND MATERIALS

Cowrie honeypot is an advanced system designed to simulate SSH (Secure Shell) and Telnet services, in order to attract hackers to enter the network and thus learn more about the attackers, in order to increase the level of protection. In this way, virus analysis can also be performed if a hacker releases a virus on the cowrie. The creators of this honeypot system are Upi Tamminen, who created Kippo - a tool written in the Python programming language, and it is a simulation of a real Linux system, which at first glance really works like a real system. In the background, the Twisted library for network programming is used, in order to properly simulate the above-mentioned protocols. So the basis of this system is Kippo together with the Twisted library. In addition, Dave Germiquet, who did unit tests and optimization, as well as Oliver Bilodeau and Ivan Korolev, who also made a huge contribution, also participated in the development. It is important to mention Florian Pelgrim, as well as Guilherme Borges, who contributed to keeping the code neat and contributing to Docker and Proxy solutions. [1] The implementation of the cowrie system has even been supported by Microsoft since October 4, 2024. When an attacker connects to this system, log files are recorded via json, but it is also possible to store these files in a sql database and send them to a remote server so that the administration team can analyze what exactly is happening. It is important to note that such a system is completely isolated from the main corporate network. [6] In this way, the methods and motives used by hackers can be learned, but also to potentially locate and catch hackers. As already mentioned, cowrie offers various possibilities, and one of the main features is that it is open-source, which means that the code is available via the Github platform, so it can be further improved, but it is also a free tool. There is also a commercial version introduced by Microsoft, where additional benefits can be obtained for a certain fee. Accordingly, it is important to note some additional powerful tools, such as Threat Intelligence. This provides insight into the growth trend of cyber attacks, as well as new techniques that are becoming more and more common every day. [2] By collecting information in this way, cybersecurity experts can significantly contribute to the protection of user systems. Based on collected data, with adequate documentation and quick response, it can significantly contribute to the protection of the system. Another important feature is the detection of previously unknown threats. This tool, when integrated with other security solutions, can catch new types of attacks before they attack the system. Also, companies can update their IDS (Intrusion Detection Systems) solutions, as well as firewall rules, based on the data collected from the cowrie system. [7]Another important advantage is the low operational risk associated with its implementation, because the system itself is completely isolated from the main one, and in this way it can be monitored more easily. [3]

Implementing Cowrie through VMWARE

The Cowrie honeypot will be installed on an Ubuntu virtual machine. As mentioned in the opening chapter, a ready-made virtual machine with Ubuntu configured will be downloaded. Using VMWare, the machine will be imported into the development environment, where cowrie will be configured. The image below shows how to import an Ubuntu virtual machine(Figure 1).

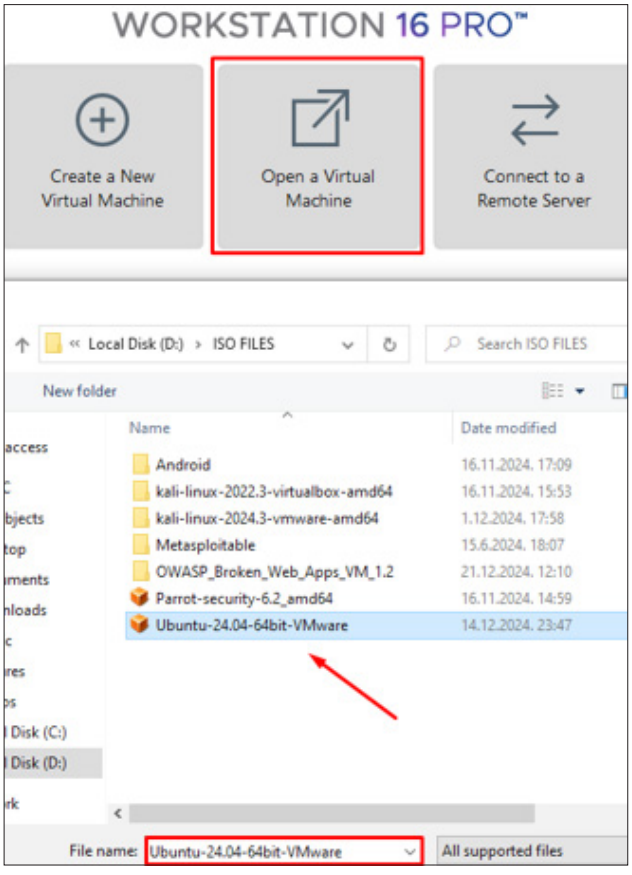


Figure 1. *Import Ubuntu virtual machine*

The name of the virtual machine within VMWare is CowrieUbuntu, for easier organization (this part is not directly visible if the machine is accessed "from the outside"). The network adapter is set to NAT network to make the virtual environment secure and isolated. Other settings such as CPU and memory are left at default, because Ubuntu can run smoothly with 2 GB of RAM and 2 VCPUs (Figure 2).

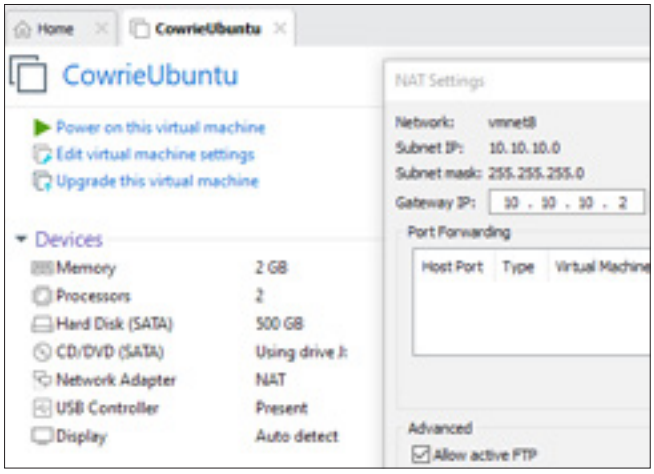


Figure 2. *Initial virtual machine setup*

Next, the virtual machine is started by clicking on "Power on this virtual machine". After updating with sudo apt update and sudo apt upgrade, it is necessary to install Git, with the command sudo apt install git, in order to be able to download cowrie from Github [4] (Figure 3)

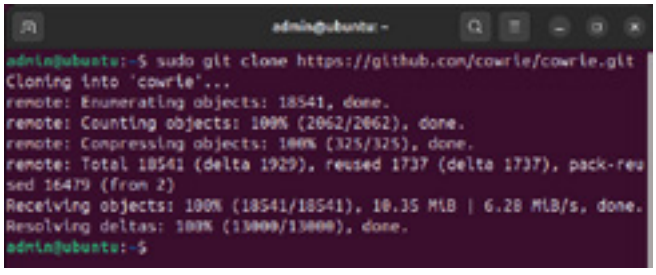


Figure 3. *Download Cowrie from Github*

The next step is to install the Python3 virtual environment. The Python virtual environment is very useful, because it provides Cowrie isolation, and thus does not affect the rest of the system. It also enables easy management and updating of the Cowrie package. Also, the security itself has been raised. The installation is done with the help of a simple command:
sudo apt install python 3.12-venv

Then, it is necessary to create a virtual environment. This is possible with the help of the following command:

python3 -m venv cowrie-env

The previously created virtual environment must then be activated. All packages installed using pip will be moved to that environment.

source cowrie-environment/bin/activate

In addition, it is necessary to install the previously mentioned pip (Python Package Installer):

sudo apt install python3-pip

Once this installation is complete, it is necessary to install everything required to run Cowrie, and the necessary installations are located within requirements.txt. [5] (Figure 4)

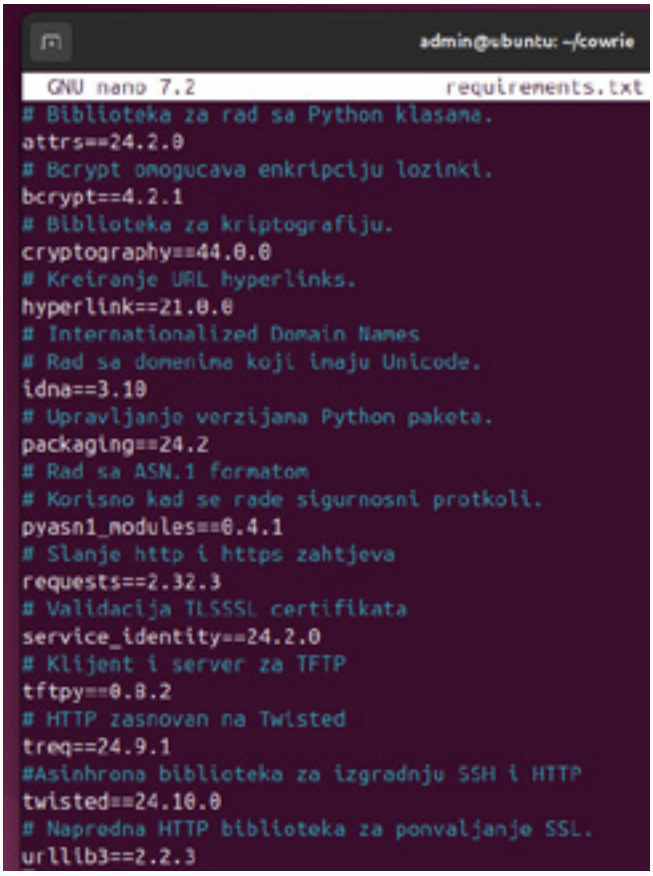


Figure 4. *The contents of the requirements.txt file*

Figure 5. *Installation of necessary tools from requirements.txt and display inside cowrie*

The Figure 5 shows the files inside cowrie. Inside honeyfs are all the necessary files that simulate a fake honeypot system. Inside etc is the main configuration file. Inside var are log files that are collected after someone tries to access the honeypot. Once this is set it is possible to start

cowrie with: *Bin/cowrie start*

Figure 6. *Running Cowrie on port 2222*

The Figure 6 shows that cowrie is properly started, and also with the help of a simple netstat command, the open port is displayed. Based on the picture, the honeypot is started, and now it is possible to monitor log files through cowrie/var/log/cowrie, and there are two types of log files stored there through .json and .log.

RESULTS

Now, as a conceptual example, a potential attack via SSH on the honeypot will be shown, using the Kali Linux machine. Kali Linux is also installed in a similar way to the Ubuntu honeypot, i.e. a ready-made virtual machine that can be downloaded online and easily implemented within VMWare Workstation. We will also take into account that the hacker has "accessed" the honeypot system, and that cowrie is located at the IP address 10.10.10.4 One of the first stages is to collect information and scan open ports, and this can be done with the help of a simple nmap tool, (Figure 7).

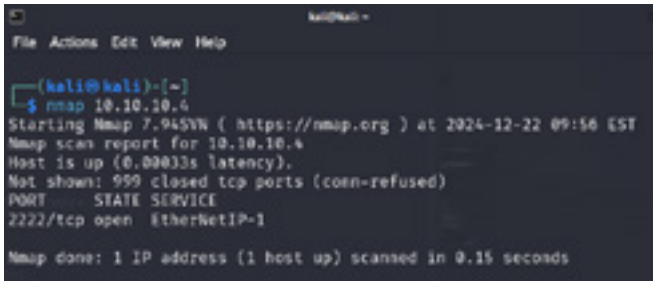


Figure 7. *Nmap scanning for open ports*

After this, the hacker will try to access using simple credentials that are already so-called well known passwords and usernames. The image below shows access to the server, which can look extremely tempting, because it really gives the impression of a real server. (Figure 8)

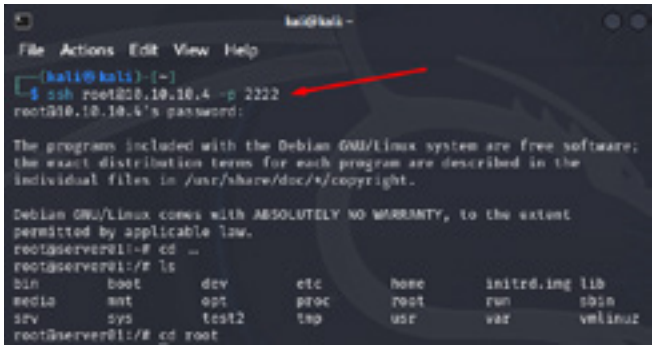


Figure 8. *Hacker access via SSH to cowrie honeypot*

The hacker is very likely not aware that he is caught in a cowrie honeypot, and that his IP address is stored inside the log file. Thus, in the log files it is possible to see the IP address, but also all the activities that the hacker tried to do, even if he tried to run a malicious script, usually of the ransomware type. (Figures 9 and 10)

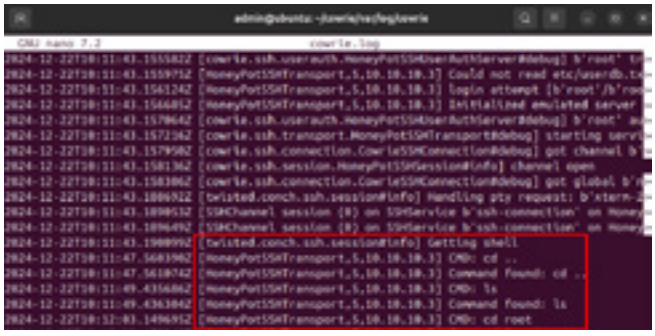


Figure 9. *View log files and all commands typed*

Figure 10. *View log files from cowrie.json file*

Given that these two types of log files exist within the cowrie directory, they are not very well struc-

tured, and are quite difficult to analyze, as can be seen in the two images above, therefore, as a contribution to this honeypot, as my contribution to this, I created a simple python script that converts the json log file into a readable output.txt file. The script and the readable log file will be shown in the Figures 11 and 12. There are various types of conversions also in the form of graphs, but the lack of an initial file, or the part that will perform the direct conversion, was noticed.

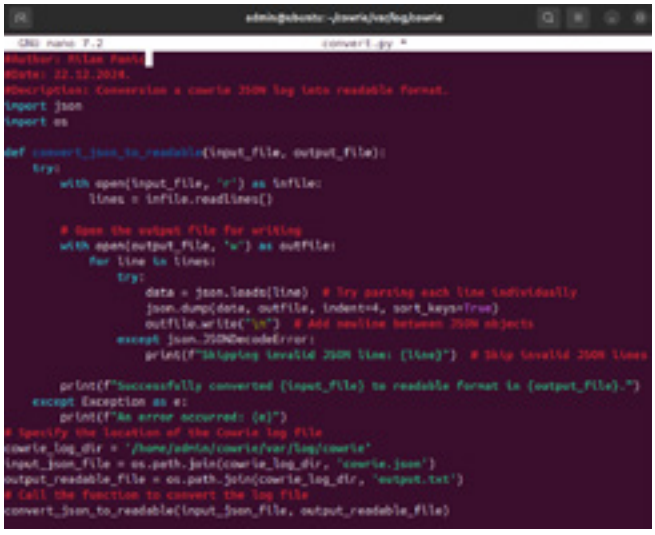


Figure 11. *Python script to convert JSON to txt*

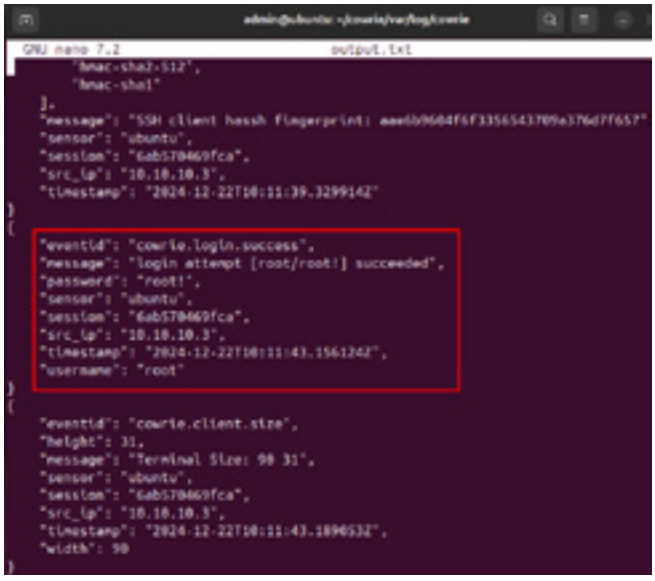


Figure 12. *Edited log file*

Discussion

The essence of the observation is a complete honeypot system, or honeynet (a system of multiple honeypots), implemented in the VMWare development environment by creating virtual machines, and can be used as an additional level of protection and defense, in order to catch and block a potential attacker in a timely manner. This research part should answer the question of whether a honeynet system can further raise the level of security and safety, in order to discover the identity of the attacker and thus disable access to the system. In addition to the functionality itself, another aspect is the material, technical and time frame that is necessary to implement such a system, whether it can also be upgraded, as well as what effect is achieved by using it. One of the main focuses is on the Cowrie honeypot system, which is designed to record Telnet and SSH connections (Figure 13). It also offers session recording capabilities. This type of honeypot is often connected to the Internet, in order to monitor additional tools, hosts and scripts that attackers use when trying to guess the password to access the system. The SSH attacker will conceptually connect to port 22, and will be redirected to our honeypot towards port 2222, as shown in the previous part of the paper.

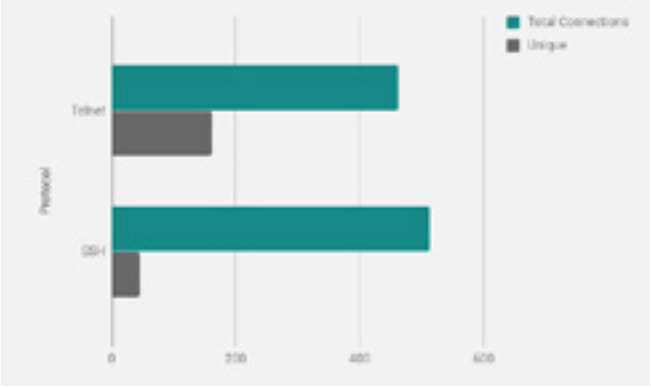


Figure 13. *Total number of attack attempts*
Source: *https://hackertarget.com/cowrie-honeypot-ubuntu/*

Conclusion

This paper explains the Cowrie honeypot system in detail, and demonstrates its functionality and effectiveness in real systems. The main goal was to understand the strategy of this interesting technique, in order to draw attackers' attention to the fake system. One limitation of this research is that the implemented Cowrie honeypot system was tested exclusively within an isolated VMware NAT environment. Although such an approach significantly improves security and prevents

unintended exposure of the host infrastructure, the obtained results may differ from those observed in publicly exposed production systems connected directly to the Internet. Real-world environments may include more complex attack patterns, larger traffic volumes, and interactions with additional security mechanisms such as IDS/IPS platforms and enterprise firewalls. Companies use honeypot systems to protect the network and servers of the entire organization, and the research part conducts academic experiments on the same at universities and schools. As is already known, with the rapid growth of artificial intelligence and cyber attacks, information security is very important for all systems, because any network that is not protected in the network can easily be compromised at any time. Based on this informations, important company data can be lost, which can represent a huge loss, and it can also be very dangerous for someone else to have access to our important personal data. Therefore, this paper provides insight into the security aspects of modern information systems, and one of the main goals was to examine whether honeypot systems are easy to hack, and to check whether they are truly isolated from the rest of the network. It is evident that traditional methods in the field of cybersecurity are increasingly giving way to modern and proactive solutions that rely on concepts such as honeypot systems. In this paper, the focus was on a specific type of honeypot model that was developed with the intention of identifying and protecting potential targets of high-tech crime. Given the rapid development of information technologies, especially in the field of artificial intelligence and adaptive security solutions, it is expected that such systems will become increasingly advanced, efficient.

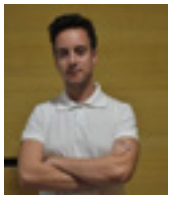
Acknowledgements
This study did not receive any external funding. The authors are fully responsible for the content of this article.

REFERENCES

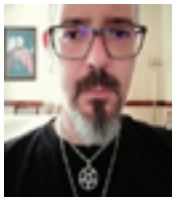
[1] A. Sardana, Honeypots - a new paradigm to information security, Science Enfield USA, 2012.
[2] S. Holder, Honeypots for Windowds, Dublin: Apress, 2014.
[3] M. Nawrocki, M. Wahlisch, T. C. Schmidt and J. Schonfelder, "A Survey on Honeypot Software and Data Analysis," *ACM Computing Surveys,* vol. 56, no. 2, 2023.
[4] O. M. Youssef and A. Almulhem, "Advanced Honeypot Architectures for Modern Cybersecurity," *IEEE Access,* vol. 12, pp. 11245-11260, 2024.
[5] "https://github.com/cowrie/cowrie," [Online]. [Accessed 25 2 2026].
[6] "https://hackertarget.com/cowrie-honeypot-ubuntu/," [Online]. [Accessed 20 2 2026].
[7] H. Jahankhani, Cyber Criminology, London: Springer Nature, 2018.

Received: March 17, 2026
Accepted: April 30, 2026

ABOUT THE AUTHORS



Milan Panić was born in Gradiška in 1999. He earned the title of Bachelor of Computer Science and Informatics Engineer in the field of computer security and information protection in July 2023 at the Pan-European University "Apeiron" Banja Luka as a top student with an average of 9.48. He completed his Master's studies in December 2025 with a grade point average of 10.00. Since 2021, he has been employed at the Pan-European University Apeiron in Banja Luka as a system administrator, and since 2024 as an assistant in the courses of computer forensics and mobile forensic. He holds a certificate from ISC2 (CC), which confirms his understanding of key principles in the field of cybersecurity, including network security, data protection and risk management.



Nemanja D. Maček is a senior lecturer at the School of Electrical and Computer Engineering and the full professor at the Faculty of Social Sciences. He also works, as a contractor, for the SECIT Security Consulting and as a senior AI researcher and scientific advisor for several companies. Nemanja graduated from University of Novi Sad in 2006 and received Advanced Security Systems PhD from Singidunum University in 2013. The research of Prof. Maček involves machine learning, pattern recognition and natu- ral language processing applied to information security, as well as biometric systems, cryptology and designing novel security mechanisms.

FOR CITATION

Milan Panić, Nemanja Maček, Implementation of Cowrie Honeypot System and Improvement of Log Analysis, *JITA – Journal of Information Technology and Applications, Banja Luka*, Pan-Europien University APEIRON, Banja Luka, Republika Srpska, Bosna i Hercegovina, JITA 16(2026)1:69-74, (UDC: 37.016::003-028.31), (DOI: 10.7251/JIT2601069P), Volume 16, Number 1, Banja Luka, June (1-76), ISSN 2232-9625 (print), ISSN 2233-0194 (online), UDC 004