Journal scientific and applied research, vol. 24, 2023
International Journal

ISSN 1314-6289 (Print)
ISSN 2815-4622 (Online)

Original Contribution

USING MODIFIED SNIFFER SCRIPTS, IMPLEMENTING LINEAR ALGORITHMS FOR DETECTION OF NETWORK PORT SCAN ATTACKS IN LINUX BASED OPERATING SYSTEMS

Petar Kr. Boyanov

COMMUNICATION AND COMPUTER TECHNOLOGIES, FACULTY OF TECHNICAL SCIENCES, KONSTANTIN PRESLAVSKY UNIVERSITY OF SHUMEN, SHUMEN 9712,115, UNIVERSITETSKA STR, E-MAIL: peshoaikido@gmail.com

ABSTRACT:
In this scientific paper modified sniffer scripts, implementing linear algorithms for detection of network port scan attacks in Linux based operating systems are presented.

KEY WORDS: *Analysis, Connection, Linux, Modified script, Monitoring, Ports, Python3, Scanning, Security, Sniffer, Traffic.*

1. Introduction

The monitoring and analyzing of network traffic as well as the detection of network port scans are essential to maintaining a well-protected host state. The main aim of network sniffers is to scan the local computer network for active hosts in order to capture network packets. Once the network packets are intercepted, the malicious users will be able to analyze and decode the information in order to find usernames, passwords, credit card numbers and details, personal information and more [7,13,14,20,21,22,23,24,25].

The purpose of the modified sniffer scripts is primarily to detect the host performing network scanning attacks and separately to determine exactly which ports have been scanned on the victim host. The programming and using of these modified scripts through the Python language ensures security on the part of the host user because no agents in the program code are presented. These agents as malicious code should be understood and in most cases they acts as a Trojans or backdoors [1,4,6,8,9,11,12,15].

The quickly discovering the perpetrator's IP address, as well as which the scanned port numbers are, allows the victim host to implement security measures and mechanisms, such as blocking the scanning host's IP address and placing the corresponding ports in a closed or filtered state. It is desirable that

the scripts must listen on every wired and wireless network interface of the victim host so that any network scanning attempts can be detected [13,14,20,22,24,25]. The conducting network scans that aim to capture and decode important and confidential information without the host's permission is considered as a crime and, if proven, is punishable to the full extent of the law of the respective country [2,3,4,5,7,11,12,17,18,19]. Everything illustrated and explained in this paper is for research work and educational purposes and the author is not responsible in cases of abuse.

2. Experiment

The scientific experiments and research works in this paper in a specialized computer network laboratory in the Faculty of Technical Sciences of the Konstantin Preslavsky University of Shumen is made. In this paper linear algorithms for detection of network port scan attacks are suggested. These algorithms are respectively designed to operate on Linux based operating systems. In this regard, fundamentally new approaches for algorithmization of activities related to network port scanning are developed.

The Python programming language has various module libraries for detection of network port scan attacks and thus the performance of modified sniffer scripts for Linux based operating systems implementing linear algorithms for detection of various types of network port scan attacks are presented.

The operation of the first modified sniffer script implementing a linear algorithm for detection of network port scan attacks in Linux based operating systems involves the following basic steps [2,3,4,5,6,7,8,9,10,16,14,19]:

1. Specifying the full path to Python.
2. Loading required modules and libraries – threating, datetime, socket, struct, time, ctypes, sys, sniff, os and queue.
3. Determining the number of detected ports.
4. Packet data unit recognition.
5. Remembering the scanned ports and current time in UNIX format.
6. Checking for the number of the scanned ports.
7. Writing the time of each network scan.
8. Delete the scanned ports and IP addresses that do not fall within the specified time.
9. Print the detected port scanner from host with IPv4 address.
10. Print the started network port capture.
11. Print the following found scanned ports.
12. Terminating the script with the key combination (Ctrl+C).

The flowchart of the first modified sniffer script, implementing a linear algorithm for detection of network port scan attacks in Linux based operating systems on fig. 1 is illustrated.

The second linear algorithm for detection of network port scan attacks in Linux based operating systems is almost the same as the previous one with the difference that it only detects the IPv4 address of the scanning host and displays the IPv4 address of the scanned host. This algorithm shows exactly how many port scan requests have been performed. The time of each request at the end of the line after the IP address of the scanned host is written. The flowchart of the second modified sniffer script, implementing a linear algorithm for detection of network port scan attacks in Linux based operating systems on fig. 2 is presented.

The scientific research using the software environment for virtualization of operating systems - VMware Workstation 12 12.5.1 build-4542065 is carried out in order to scan and detect open ports on active hosts in the computer network. The virtual installed operating system for the two hosts is respectively Linux pesho 6.0.0-kali6-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.0.12-1kali1 (2022-12-19) x86_64 GNU/Linux. The both scripts do not have any malware embedded in it, and thus a network specialists, network pentesters or users can use it for performing detection of scan ports without having to worry about being infected with viruses, worms, backdoors and rootkits.

The aim of using Linux virtual machines is to cut off physical access to both the underlying installed operating system and direct access with the hardware of the hosted computing machine. There is always a risk of compromising the underlying operating system on which the VMware environment is installed. In this regard, performing regular backups to external media completely solves the problem.

After that it follows scanning and discovering both the physical MAC addresses and the logical network IP addresses of the hosts on a corresponding computer network. A special command is used to scan the network number 192.168.80.0 in order to find all active hosts. Since the netmask is 24-bit, then the maximum number of active hosts is 254. Only two active hosts are found. The attacking host has an IPv4 address 192.168.80.132 and the victim host has an IPv4 address 192.168.80.130.

The both modified sniffer scripts, implementing a linear algorithm for detection of network port scans attacks on the host (192.168.80.130) with Kali Linux virtual operating system are executed. On the attacking host (192.168.80.132) a modified script for Linux based operating systems using a linear algorithm for network port scanning is executed (shown on fig. 3 and 4).

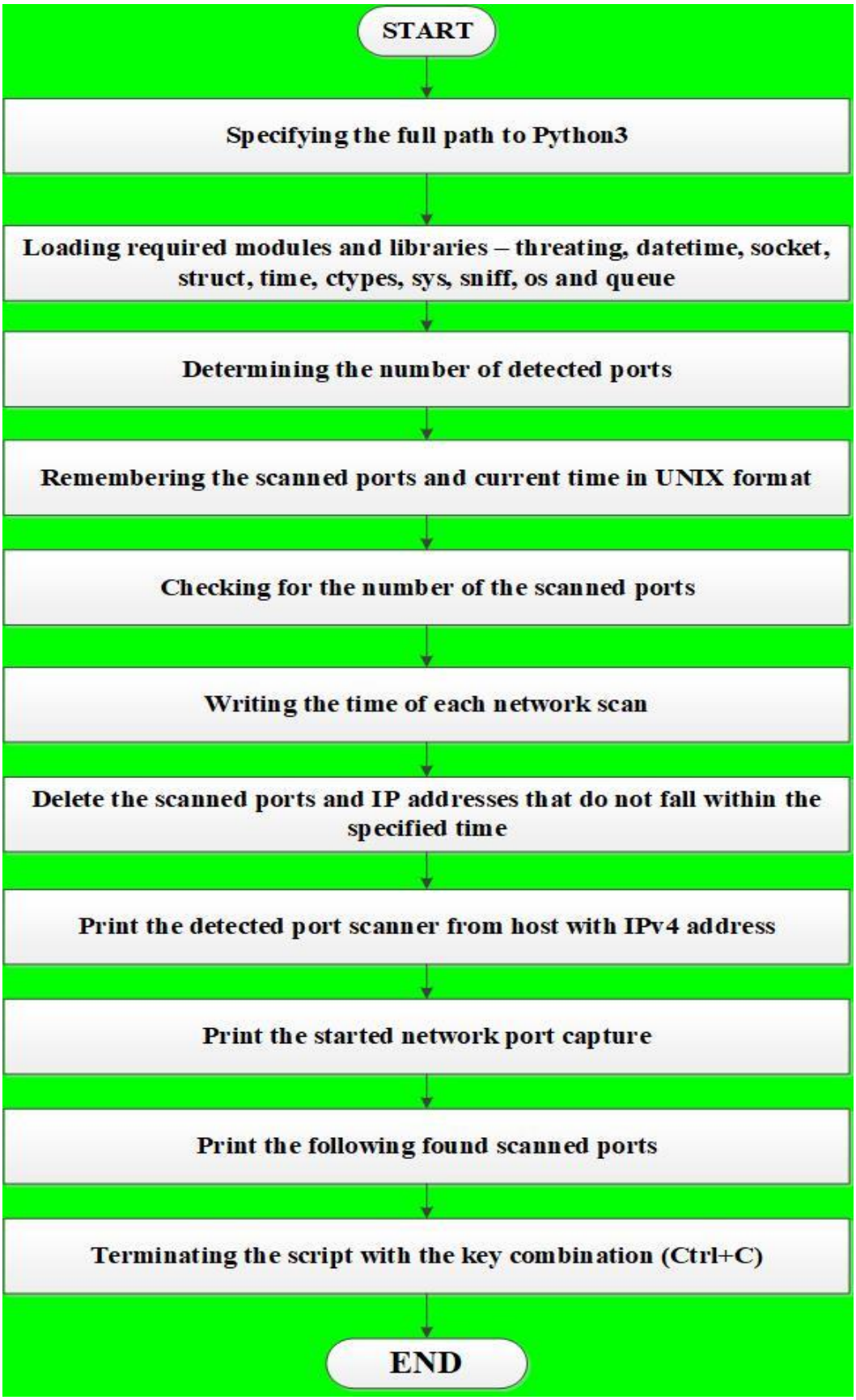

Fig. 1. Flowchart of the first modified sniffer script, implementing a linear algorithm for detection of network port scans attacks

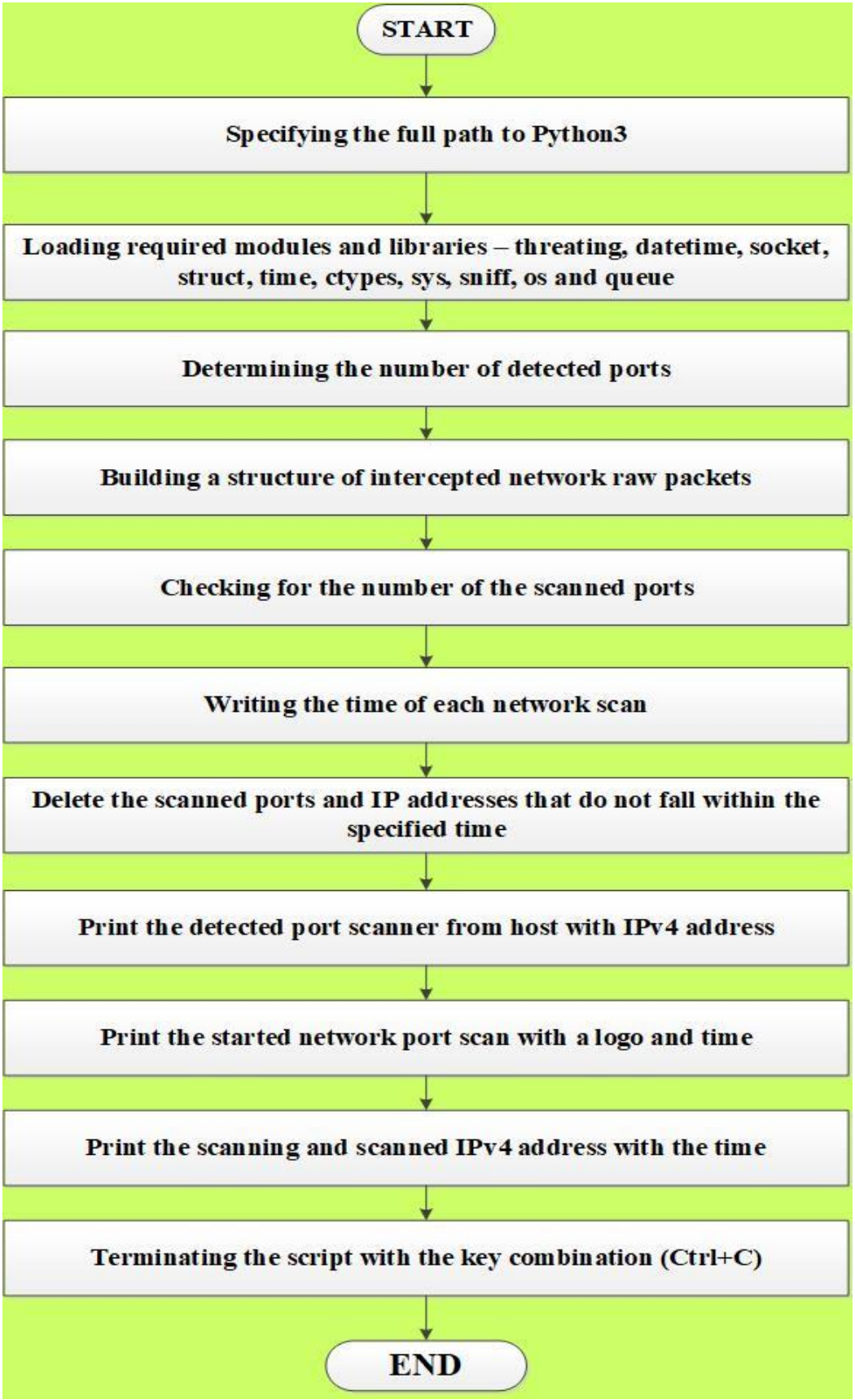

Fig. 2. Flowchart of the second modified sniffer script, implementing a linear algorithm for detection of network port scans attacks

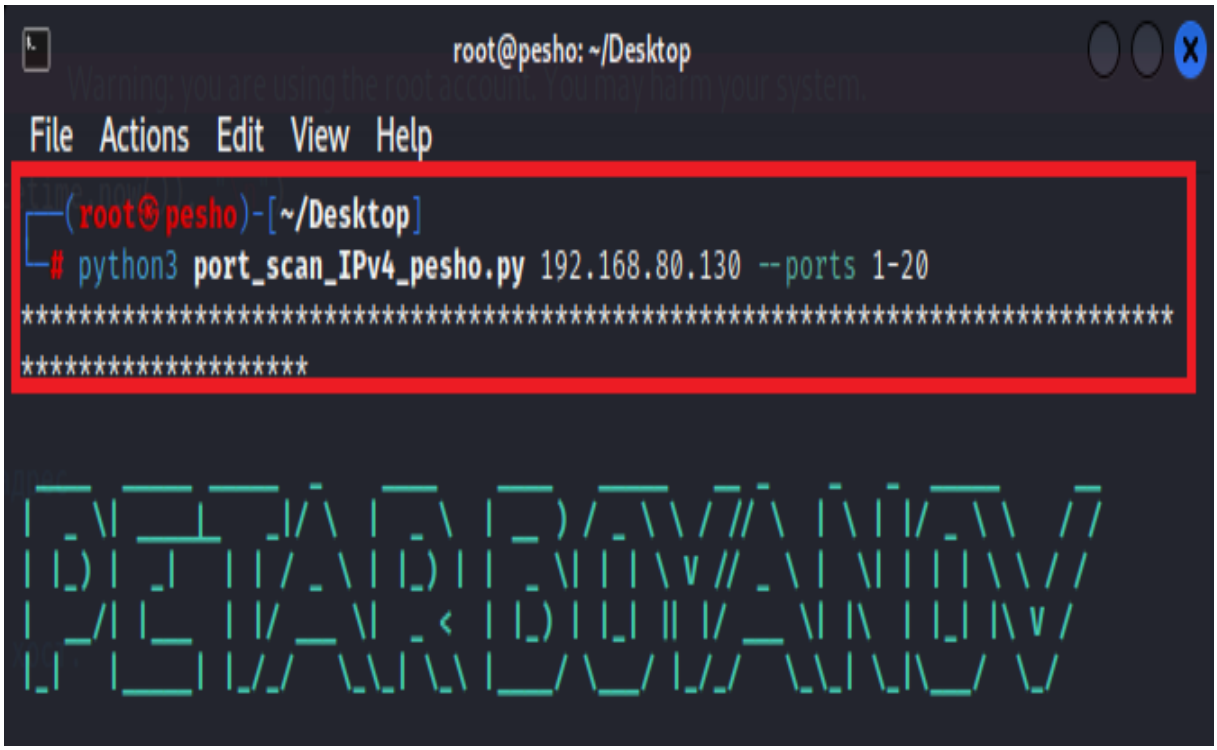

Fig. 3. The execution of the modified script "python3 port_scan_IPv4_pesho.py 192.168.80.130 --ports 1-20"

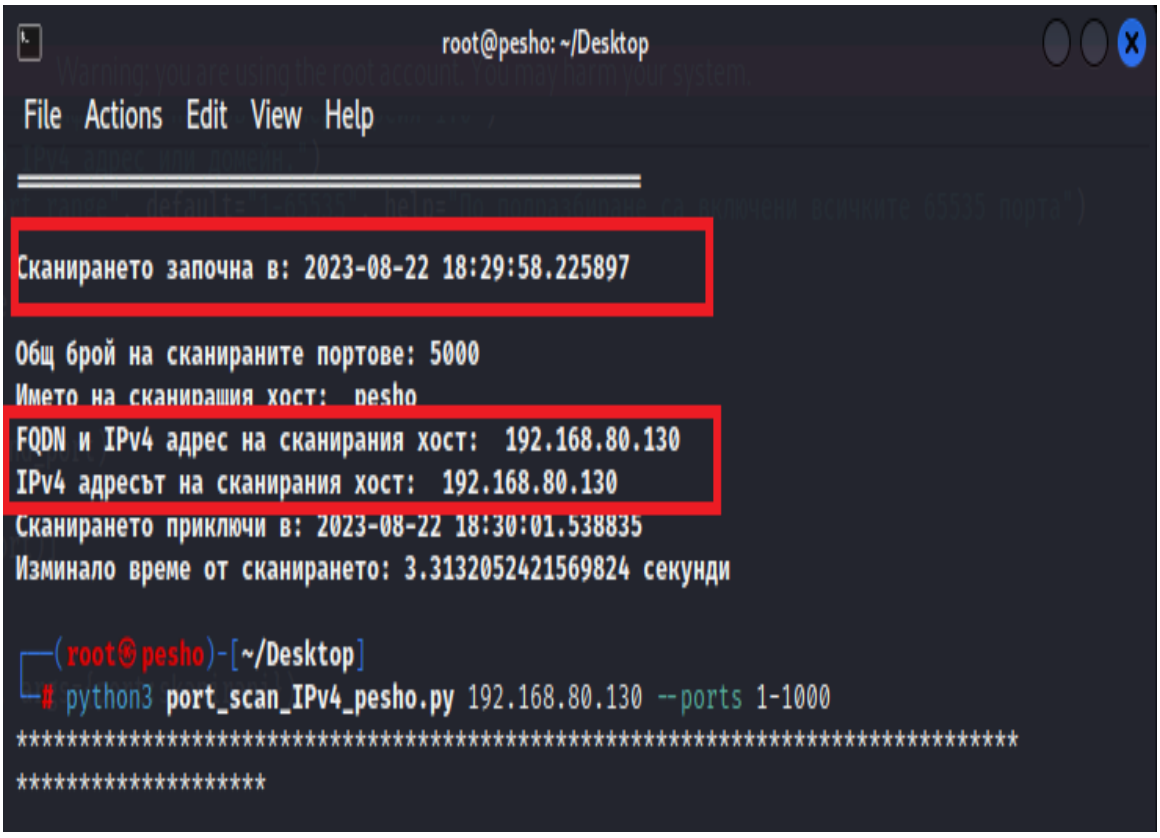

Fig. 4. The execution of the modified script "python3 port_scan_IPv4_pesho.py 192.168.80.130 --ports 1-5000"

3. Results

Figures 3 and 4 show the results obtained after performing a network port scan on the host with address 192.168.80.132 for the first 20 and 1000 ports using the commands "python3 port_scan_IPv4_pesho.py 192.168.80.129 --ports 1-20" and "python3 port_scan_IPv4_pesho.py 192.168.80.130 --ports 1-5000". The obtained results of the executed first modified sniffer script on fig. 5 are presented. The obtained results of the executed second modified sniffer script on fig. 6 are illustrated.

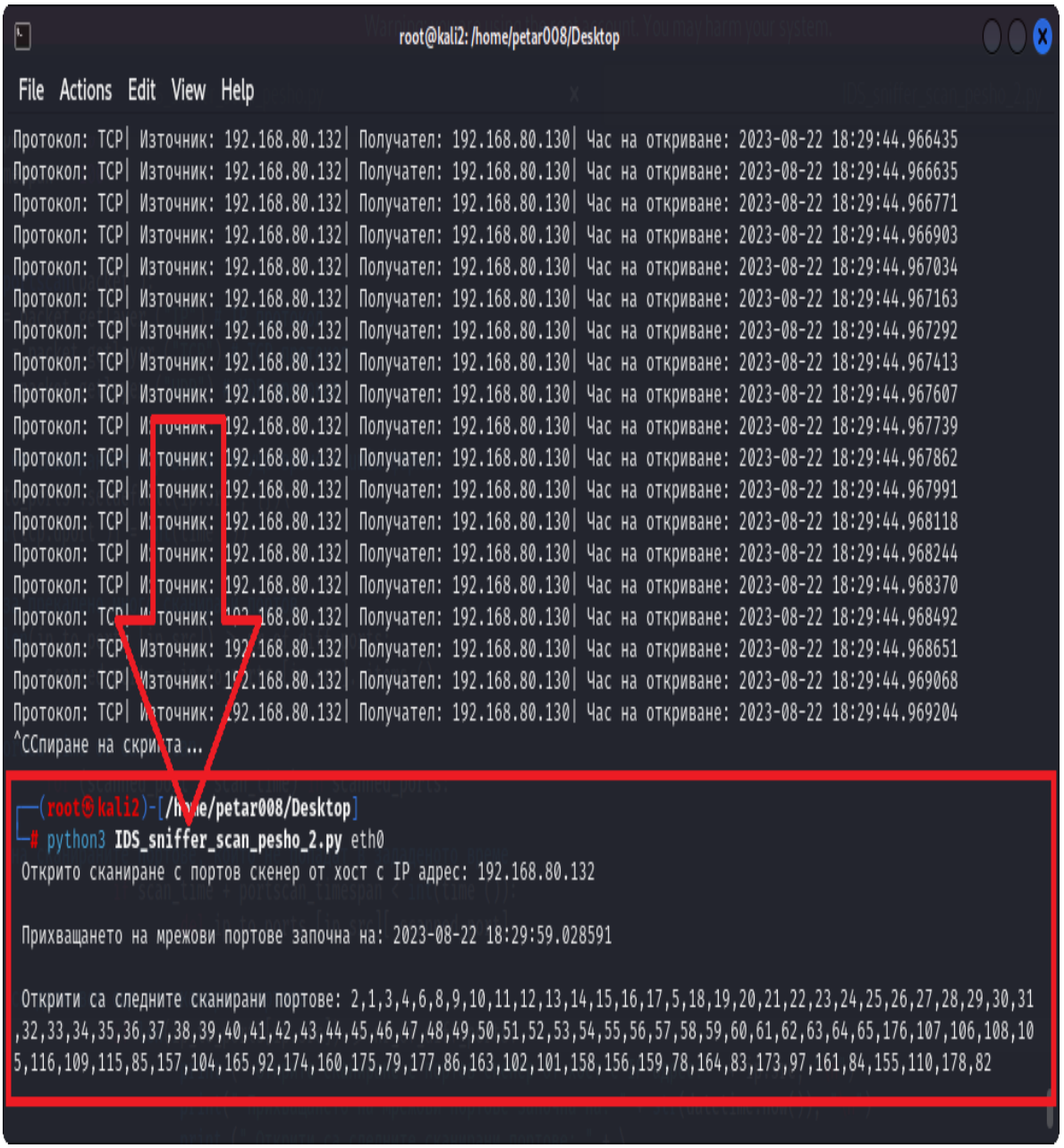

Fig. 5. The first executed detection sniffer script

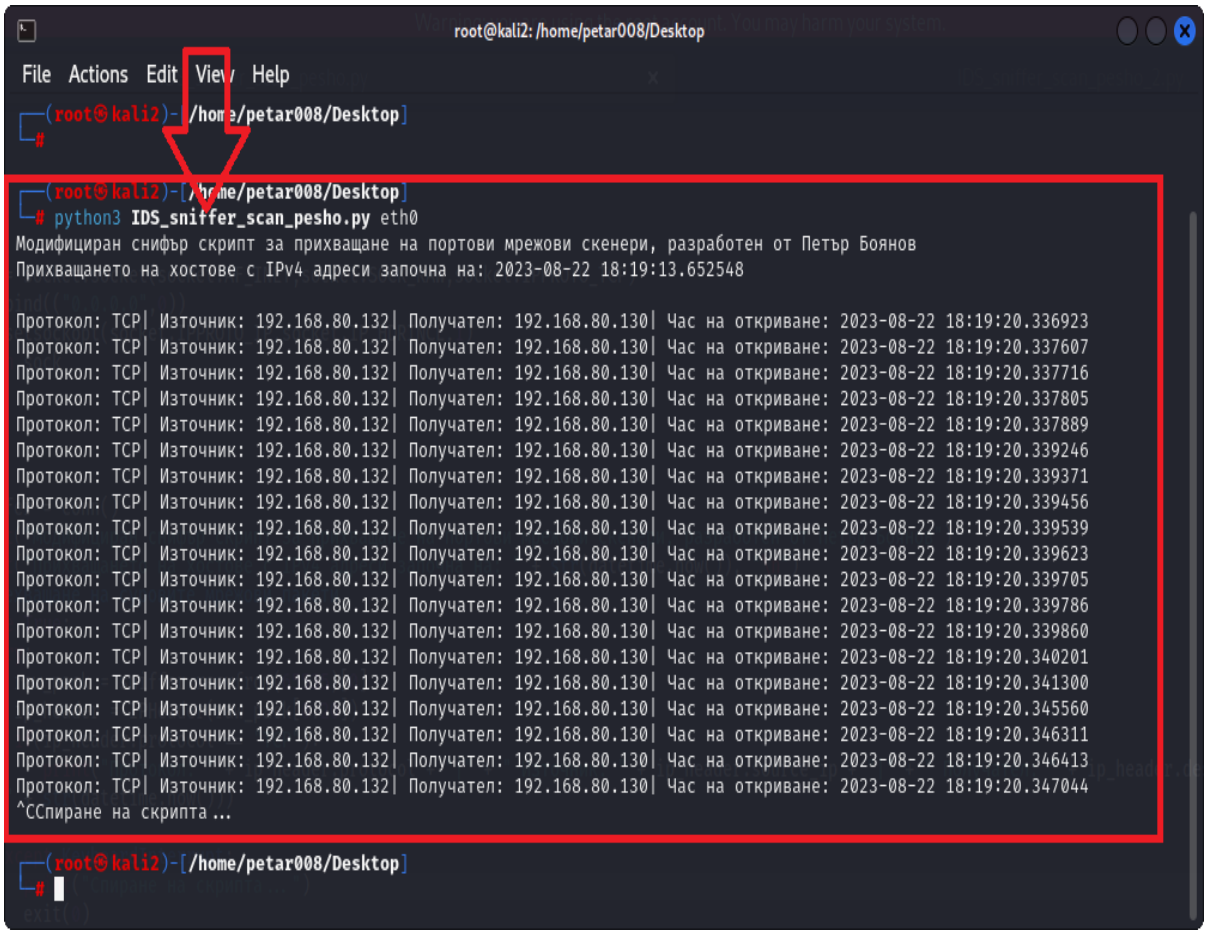

Fig. 6. The second executed detection sniffer script

The results of the conducted scientific research show that the modified sniffer scripts are able to intercept all network packets sent to the victim host with IPv4 address 192.168.80.130. The detection time of each port scan is almost instantaneous.

3. Conclusion

The rapid detection of network scanning attacks is of primary importance for any security engineers and pentesters whose tasks are to ensure the protection of the information resources of the determined computer system. Thanks to the both modified detection sniffer scripts, it is possible to catch any network port scans and thus the white hats can block the IPv4 address of the malicious user in time and quickly. In this regard the exceptionally well-equipped laboratories at the Faculty of Technical Sciences at the Konstantin Preslavsky University of Shumen give great opportunities to students majoring in "Communication and Information Systems", "Computer Technologies in Automated Manufacturing" and "Signal Security Systems and Technologies" to gain extensive theoretical and practical experience in the detection of various types of network scan attacks.

Acknowledgments

This scientific article under project number RD-08-153/28.02.2023 „Programming of PIC and AVR microcontrollers“ is funded.

References:

[1] Arkamburge, Python Quick Basics and Advacnced programming Guide for Dummies and beginners on coding in computer science design using tricks with javascipt along learning to network and hack quickly: Guide 2 code. Independently published, ISBN-13 2021, 979-8768745837, p. 193.

[2] Arnodlz, J., Python for Hackers and Pentesters full guides: you'll explore the darker side of Python's capabilities - writing network sniffers, stealing email ... fuzzers, infecting virtual machines.... Independently published, ISBN-13 2021, 979-8712511532, p. 185.

[3] Asrodia, Pallavi, and Hemlata Patel. "Network traffic analysis using packet sniffer." International journal of engineering research and applications 2, no. 3 (2012): 854-856.

[4] Ballmann, B., Understanding Network Hacks: Attack and Defense with Python 3 2nd ed. Springer, ISBN-10 3662621592, ISBN-13 978-3662621592, 2022, p. 236.

[5] Ballmann, B., Network Hacks - Intensivkurs: Angriff und Verteidigung mit Python 3 (German Edition). Springer, ISBN-10 3662616351, ISBN-13 2020, 978-3662616352, p. 237.

[6] Barth, A., Caballero, J., & Song, D. (2009, May). Secure content sniffing for web browsers, or how to stop papers from reviewing themselves. In 2009 30th IEEE Symposium on Security and Privacy (pp. 360-371). IEEE.

[7] Boyanov, P., Implementation of modified script for Linux based operating systems using a linear algorithm for network port scanning. A refereed Journal Scientific and Applied Research, Konstantin Preslavsky University Press, Vol. 23, Shumen, 2022, ISSN 1314-6289 (Print), ISSN 2815-4622 (Online), pp. 48-59, DOI: https://doi.org/10.46687/jsar.v23i1.353.

[8] Boyanov, P., A comprehensive scanning for open, closed and filtered ports in the computer systems and networks. A refereed Journal Scientific and Applied Research, Konstantin Preslavsky University Press, Vol. 23, Shumen, 2022, ISSN 1314-6289 (Print), ISSN 2815-4622 (Online), pp. 85-98, DOI: https://doi.org/10.46687/jsar.v23i1.356.

[9] Candel, J., Mastering Python for Networking and Security: Leverage the scripts and libraries of Python version 3.7 and beyond to overcome

networking and security issues, 2nd Edition. Packt Publishing, ISBN-10 1839217162, ISBN-13 2021, 978-1839217166, p. 538.
[10] Candel, J., Sarker, M., Washington, S., Learning Python Networking: A complete guide to build and deploy strong networking capabilities using Python 3.7 and Ansible, 2nd Edition. Packt Publishing, ASIN B07Q4SDBGZ, 2019, p. 492.
[11] Candel, J., Mastering Python for Networking and Security: Leverage Python scripts and libraries to overcome networking and security issues. Packt Publishing, ISBN-10 1788992512, ISBN-13 978-1788992510, 2018, p. 426.
[12] Candel, J., Hacking ético con herramientas Python (Colecciones ABG - Informática y Computación) (Spanish Edition). American Book Group, ISBN-10 168165699X, ISBN-13 978-1681656991, 2020, p. 290.
[13] Choi, B., Introduction to Python Network Automation: The First Journey 1st ed. Edition. Apress, ISBN-10 1484268059, ISBN-13 978-1484268056, 2021, p. 896.
[14] Chou, E., Kennedy, M., Whaley, M., Mastering Python Networking: Your one-stop solution to using Python for network automation, programmability, and DevOps, 3rd Edition. Packt Publishing, ISBN-10 1839214678, ISBN-13 2020, 978-1839214677, p. 576.
[15] Codings, Z., Python Machine Learning: A Beginner's Guide to Python Programming for Machine Learning and Deep Learning, Data Analysis, Algorithms and Data Science With Scikit Learn, TensorFlow, PyTorch and Keras. Independently published, ISBN-10 1696563119, ISBN-13 978-1696563116, 2019, p. 147.
[16] Codings, Z., Computer Programming And Cyber Security for Beginners: This Book Includes: Python Machine Learning, SQL, Linux, Hacking with Kali Linux, Ethical Hacking. Coding and Cybersecurity Fundamentals. Independently published, ISBN-10 1671532902, ISBN-13 978-1671532908, 2019, p. 408.
[17] David, M., Mastering Python Network Programming: learn Network programming in simple and easy steps using Python as a programming language. Independently published, ISBN-13 2021, 979-8758780589, p. 103.
[18] Elghaly, Y., Learn Penetration Testing with Python 3.x: Perform Offensive Pentesting and Prepare Red Teaming to Prevent Network Attacks and Web

Vulnerabilities (English Edition). BPB Publications, ISBN-10 9390684919, ISBN-13 2021, 978-9390684915, p. 344.
[19] Iliev, R., K. Ignatova. Cloud technologies for building data center system for defense and security. T. Tagarev et al. (eds.), Digital Transformation, Cyber Security and Resilience of Modern Societies, Studies in Big Data 84, , ISBN 978-3-030-65721-5, Springer 2020, pp. 13-24, https://doi.org/10.1007/978-3-030-65722-2.
[20] Kulshrestha, A., & Dubey, S. K. (2014). A literature reviewon sniffing attacks in computernetwork. International Journal of Advanced Engineering Research and Science (IJAERS), 1(2).
[21] Malek, M. S. A., & Amran, A. R. (2021). A Study of Packet Sniffing as an Imperative Security Solution in Cybersecurity. Journal of Engineering Technology, 9(1), 96-101.
[22] Oluwabukola, O., Oludele, A., Ogbonna, A. C., Chigozirim, A., & Amarachi, A. (2013, July). A Packet Sniffer (PSniffer) application for network security in Java. In Proceedings of the Informing Science and Information Technology Education Conference (pp. 389-400). Informing Science Institute.
[23] Pavlova, D., Dzhelepov, V., Gindev, P., Effectiveness of information security in computer systems for object and process management. 13th International traveling seminar, Modern dimensions in European education and research area. Bulgarian-Austrian cultural dialogue, 26-31 May 2019, Sofia, “ZA BUKVITE – O Pismeneh” Publishing House, vol. 7, 2019, pp. 241-249. ISSN 2367-7988.
[24] Qadeer, M. A., Iqbal, A., Zahid, M., & Siddiqui, M. R. (2010, February). Network traffic analysis and intrusion detection using packet sniffer. In 2010 Second International Conference on Communication Software and Networks (pp. 313-317). IEEE.
[25] Thakur, B. S., & Chaudhary, S. (2013). Content sniffing attack detection in client and server side: A survey. International Journal of Advanced Computer Research, 3(2), 7.