Journal scientific and applied research, vol. 23, 2022
International Journal

Original Contribution

ISSN 1314-6289

IMPLEMENTATION OF MODIFIED SCRIPT FOR LINUX BASED OPERATING SYSTEMS USING A LINEAR ALGORITHM FOR NETWORK PORT SCANNING

Petar Boyanov

DEPARTMENT OF MANAGEMENT OF SECURITY SYSTEMS, FACULTY OF TECHNICAL SCIENCES, KONSTANTIN PRESLAVSKY UNIVERSITY OF SHUMEN, SHUMEN 9712,115, UNIVERSITETSKA STR,

E-MAIL: petar.boyanov@shu.bg

ABSTRACT: *The Python programming language has various module libraries for network scanning of hosts. In this paper the performance and implementation of a modified script for Linux based operating systems using a linear algorithm for network port scanning is presented.*

KEY WORDS: *Analysis, Connection, Monitoring, Network, Ports, Python, Scanning, Services, Traffic.*

1. Introduction

The TCP or UDP port scanning is a popular reconnaissance technique used in the contemporary cyberattacks. All computer systems (hosts) connected to a network run services that listen on well-known and less-known network ports. The port scanning helps a malicious attacker to find which ports are open, closed or filtered i.e. which service can listen on the given port. Essentially, port scanning consists of sending a network raw message to each port. The type of response received indicates whether the port is in use so that it can then be further probed for weaknesses and vulnerabilities [1,3,5.7,10,22,23,24].

If the network port scanning with malicious intent is done, then the malicious cybercriminals usually prefer to remain undetected. The computer and network security applications can be configured to alert system administrators if network connection requests on a wide range of ports from a single host are detected [5,6,8,11,12,13,14]. To circumvent this option, an attacker can perform a network port scan in two modes. The first mode limits ports to a smaller target range instead of scanning all 65536 ports. The second scan mode uses scan delay techniques. By scanning ports for a much longer period of time, the

chance of detecting and preventing an intruder's actions is dramatically reduced [2,4,6,7,8,9,19,20,21]. In this scientific research, the main emphasis on the implementation of modified script for Linux based operating systems using a linear algorithm for network port scanning is placed.

2. Experiment

The experiment in a specialized computer network laboratory in the Faculty of Technical Sciences is made. In this paper a linear algorithm for network port scanning is suggested. This algorithm is respectively designed to operate on Linux based operating systems. In this regard, fundamentally new approaches for algorithmization of activities related to network port scanning is developed.

The Python programming language has various module libraries for network scanning of hosts and thus the performance of a modified script for Linux based operating systems implementing a linear algorithm for network port scanning is presented.

The operation of the modified script implementing a linear algorithm for network port scanning for Linux based operating systems involves the following basic steps:

1. Specifying the full path to Python.
2. Loading required modules and libraries.
3. Configure network scan start time - start_time=date.time().
4. Defining the function to get the version of the services and protocols used (get_banner).
5. Defining the colors used in the network port scanner.
6. Defining the name and version of the modified network port scanner.
7. Refinement of the IP address conversion function - check_ip.
8. Checking the IP address of the scanned host and resolving a domain name into an IP address.
9. Defining the number of scan threads - N_THREADS and the thread queue.
11. Defining the port scan function with the global variable host.
12. Defining the function scan_thread().
13. Initialization of each thread and queuing each port to begin scanning.
14. Waiting for threads to finish scanning ports.
15. Configuring the scan parameters and getting help information.
16. Displaying the total number of ports scanned and displays the name of the scanning host.
17. Getting information about the scanned host's FQDN and IPv4 address and display only the IPv4 address of the scanned host.
18. Displaying the time when the scan is completed.
19. Displaying the elapsed time after the scanning process has started.

20. Displaying detailed information about only found open ports on the victim host.

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 systems for network research are respectively:

- Windows 10 Pro x64;
- Windows 7 Professional x64;
- Kali Linux 2022.2 amd64.

The network port scanner does not have any malware embedded in it, and thus a specialists or users can use it for performing host scan without having to worry about being infected with viruses and worms.

The purpose of using 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. All installed virtual operating systems for scientific research on fig. 1 are showed.

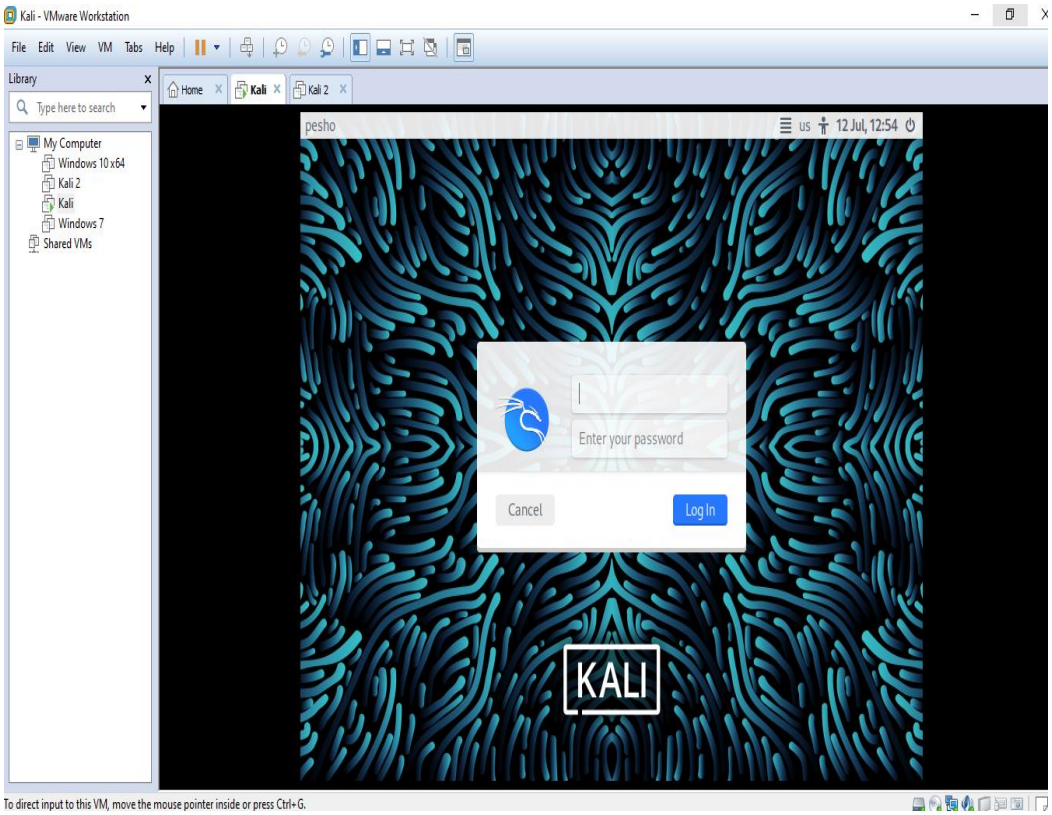

Fig. 1. All installed virtual operating systems for scientific research

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. The whole number of found active hosts on fig. 2 is showed. The host with IPv4 address 192.168.80.129 is running a Windows 10 virtual operating system, and the host with IPv4 address 192.168.80.130 is running a Kali Linux virtual operating system. The modified script on the host (192.168.80.130) with Kali Linux virtual operating system is executed.

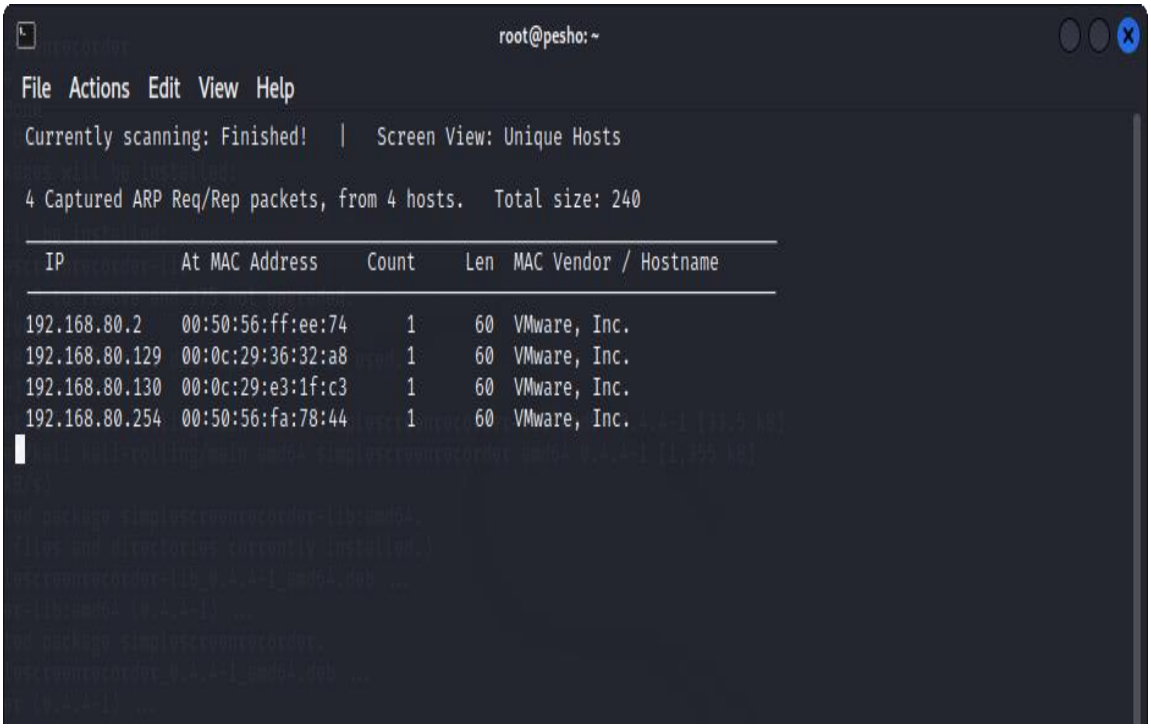

Fig. 2. Active found hosts in the local computer network

3. Results

Figures 3 and 4 show the results obtained after performing a network port scan on the host with address 192.168.80.129 for the first 1001 ports using the command "python3 port_scan_IPv4_pesho.py 192.168.80.129 --ports 1-1001". It is found that 5 open ports are detected, and their numbers are 21, 25, 106, 110 and 143. Detailed information about the started service of each port is revealed separately. Thanks to this information, the malicious perpetrator can use the most correct exploit to perform unauthorized and unsanctioned access to the resources of the victim host. The total time of the port scan is 1.36 seconds. The obtained detailed information about the started services on the detected open ports is accordingly:

- Open port 21 - 220-FileZilla Server version 0.9.41 beta 220-written by Tim Kosse (Tim.Kossefi)gmx.de).

- Open port 25 - 220 localhost ESMTP server ready.
- Open port 106 - 200 localhost MercuryW PopPass server ready.
- Open port 110 - <214197781.5192@localhost>, POP3 server ready.
- Open 143 - localhost IMAP4revl Mercury/32 V4.62 server ready.

The presented modified script for Linux based operating systems in Bulgarian Defense Institute can be used in order to be detected open unprotected network ports. In relation to this the chief information security officers will be able to take timely measures to implement protective mechanisms and policies for the protection of the information resources containing critical and confidential information about data centers in defense and security, jamming devices, bullets, ammunitions, projectiles, rocket motors and ballistic materials [3,7,11,12,15,16,17,18,19,20,25].

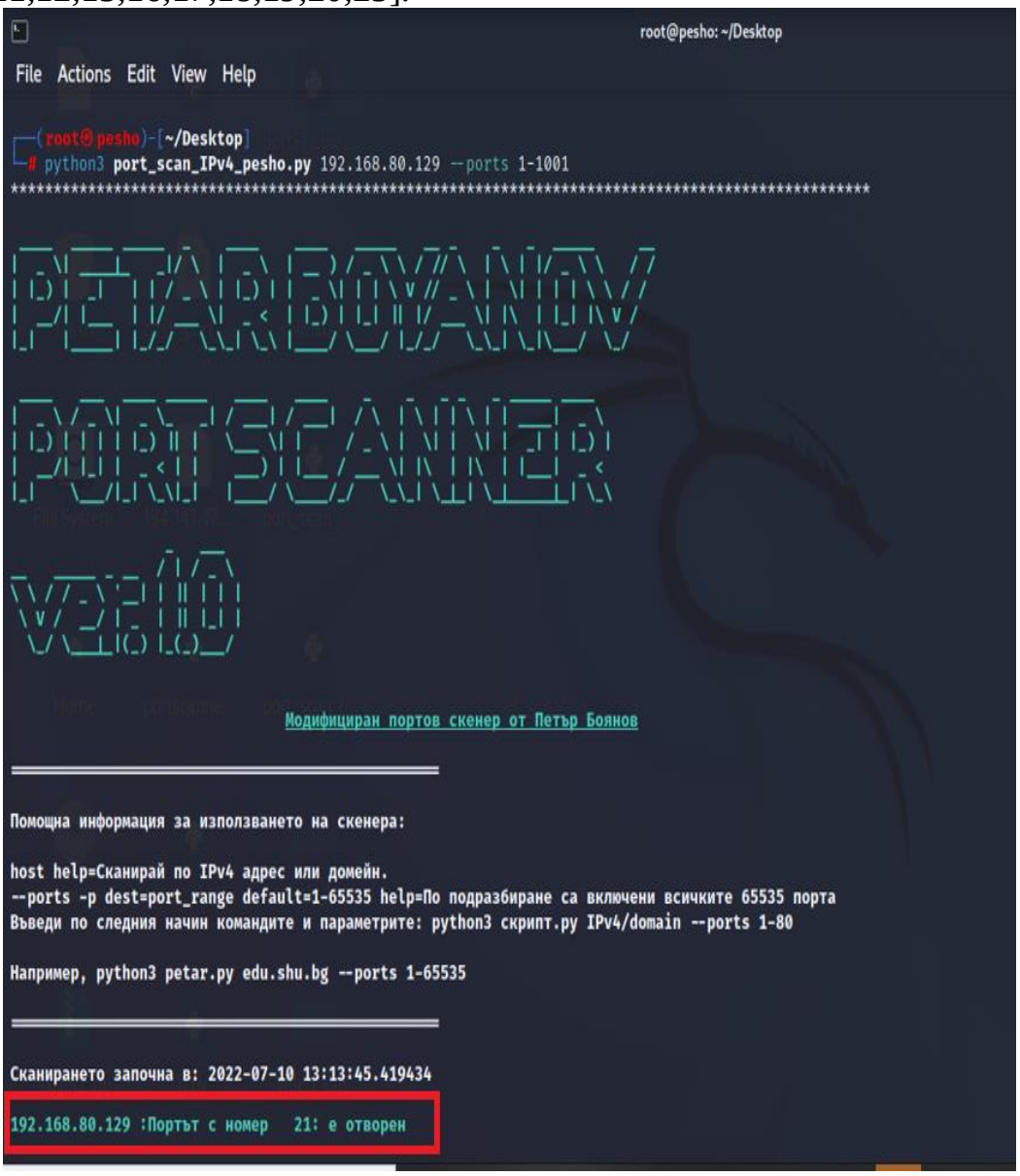

Fig. 3. The execution of the modified script "python3 port_scan_IPv4_pesho.py 192.168.80.129 --ports 1-1001"

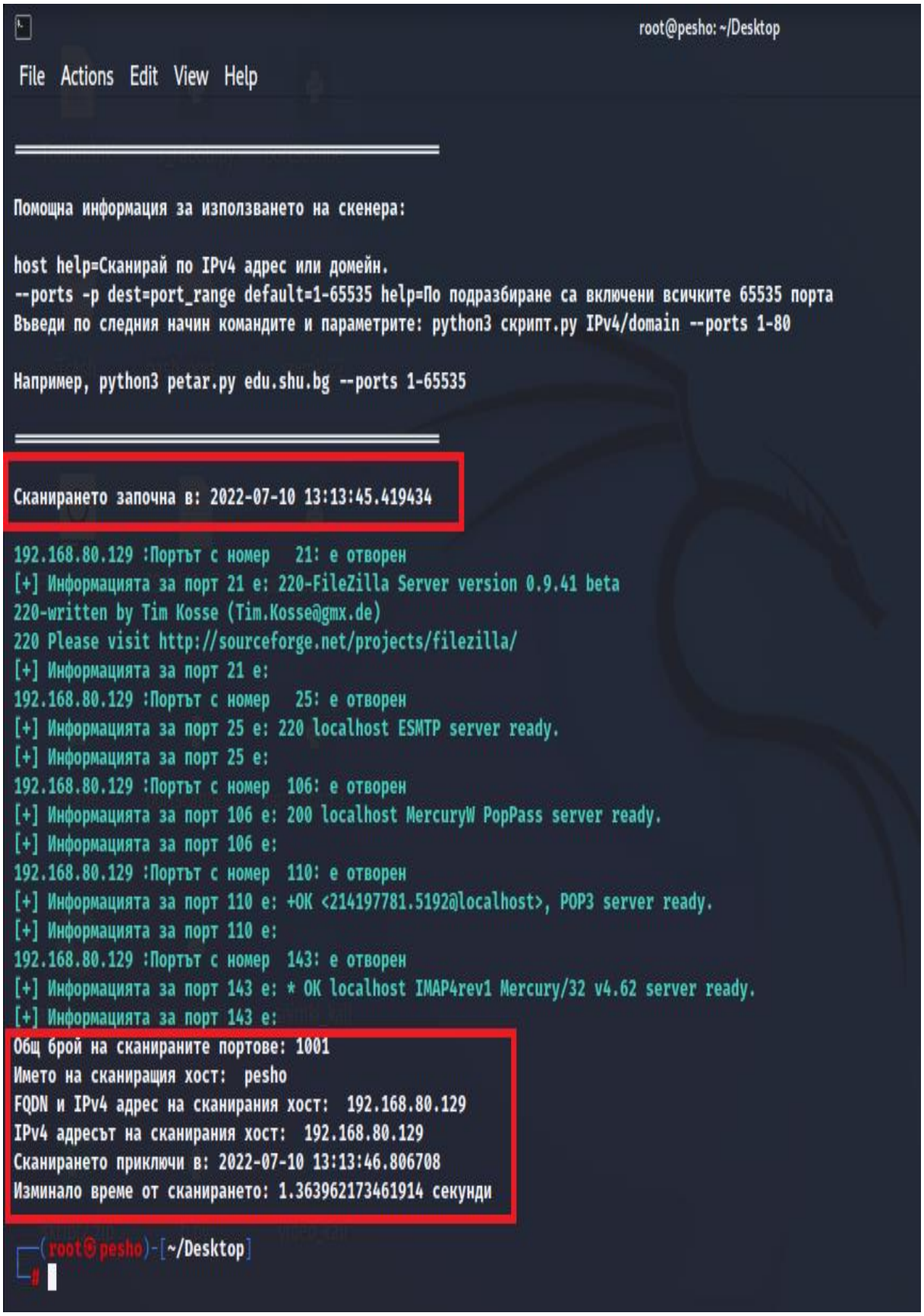

Fig. 4. The received results of the performed port scan

Separately, the modified script reveals the fully qualified domain name (FQDN), which in this case is small-sites-tmp.shu.bg and the public IPv4 address of the subdomain jsar.ftn.shu.bg. The subdomain IP address is 194.141.47.8. The total port scan time for 22.377 seconds is performed. An additional port network scan process follows on the public IPv4 address – 194.141.47.8 to uncover additional open ports. The direct scan by IP address gives the best and most detailed results for all open ports on the host. From the obtained results in figures 5 and 6, it is found that a total of 3 open ports are detected out of a total of 100 scanned. Again, thanks to the information obtained, the malicious perpetrator now has a large selection of exploits to compromise the corresponding detected open ports.

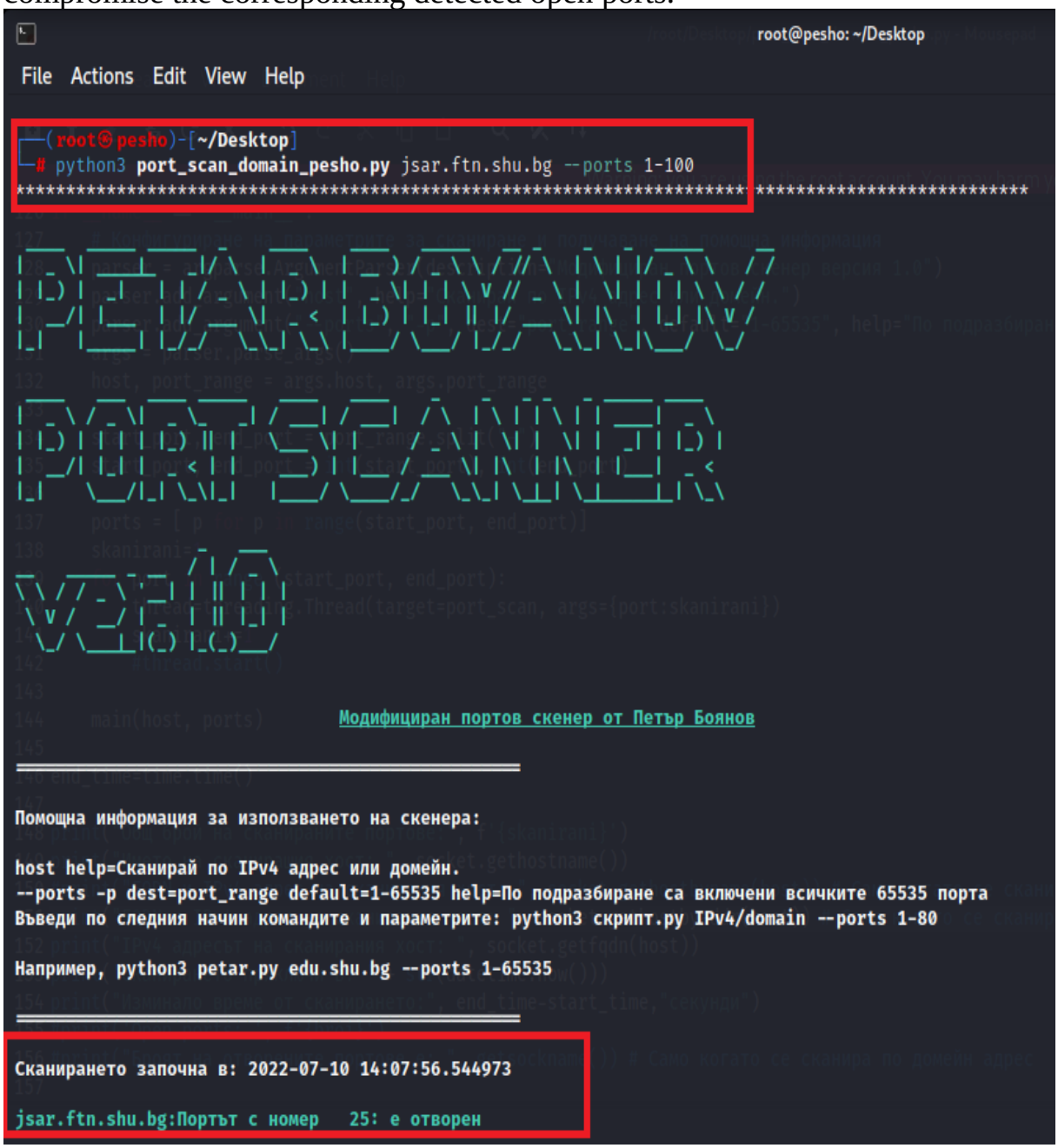

Fig. 5. The port scanning of subdomain jsar.ftn.shu.bg for the first 100 ports

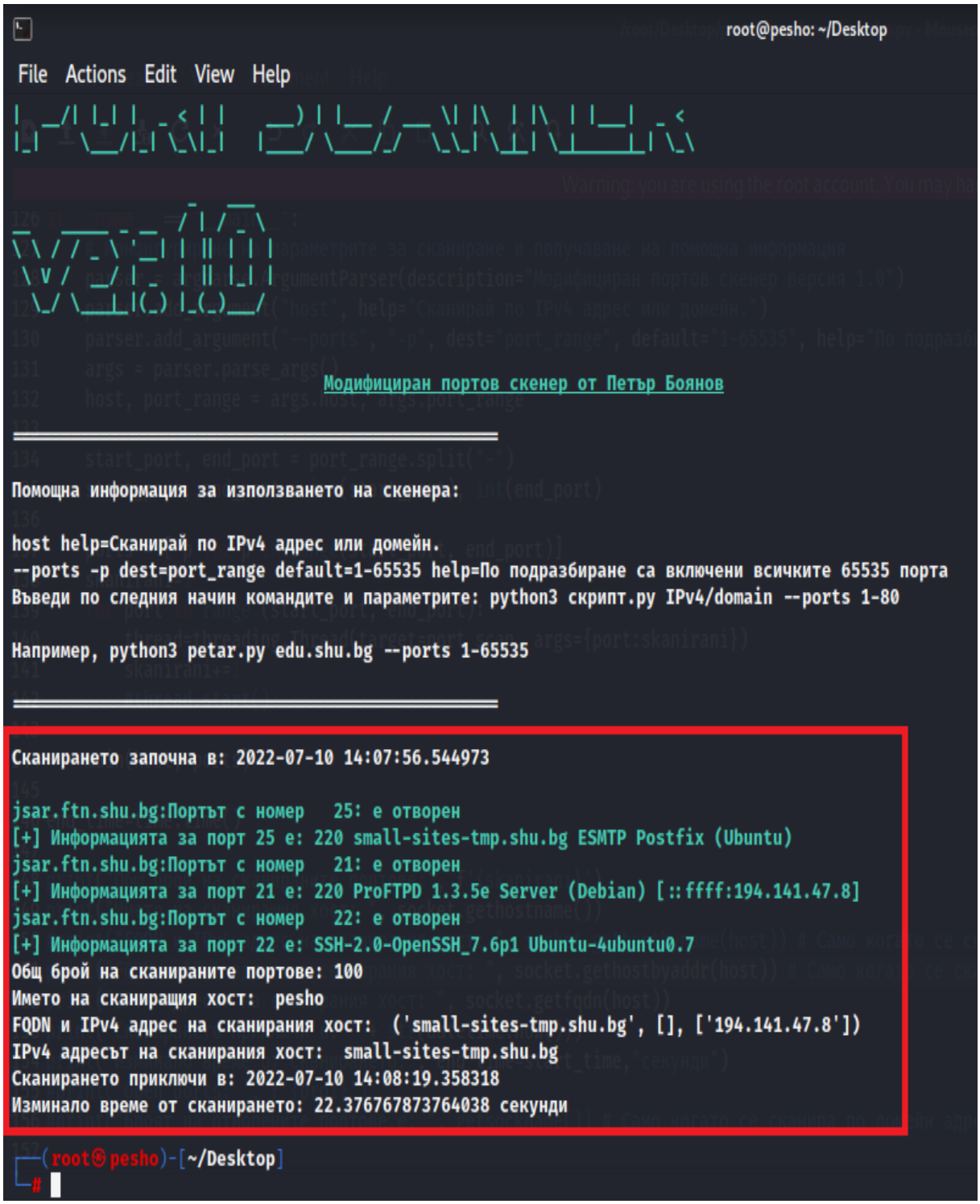

Fig. 6. The obtained results from the execution of the command "python3 port_scan_domain_pesho.py jsar.ftn.shu.bg --ports 1-100"

The results of the conducted scientific research show that the higher the number of threads used, the higher the probability of an open port being detected. It is found that when using 550 threads, more open network ports are found on the host and this is shown on fig. 7.

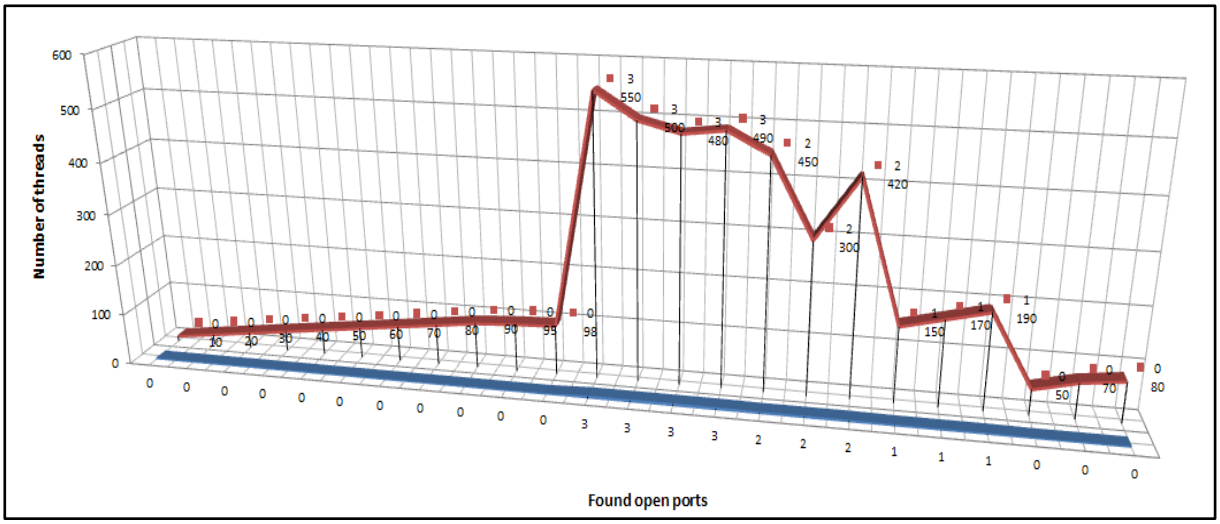

Fig. 7. The statistical processing of the obtained results of the conducted scientific research

ATTENTION: The scientific experiments and research works in this paper in a specialized computer laboratories at the Faculty of Technical Sciences of the Konstantin Preslavsky University of Shumen are made. Everything illustrated and explained in this paper is for research work and educational purposes and the authors are not responsible in cases of abuse.

3. Conclusion

The malicious attackers penetrate computer networks and systems for various purposes. Therefore, it is important to understand how malicious actors attack and exploit computer systems and what the likely reasons for carrying out these cyberattacks are. The system administrators and information security professionals must protect their infrastructure from various types of malicious software tools - exploits, by knowing the adversary who seeks to use this infrastructure for illegal activities. Thus 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 analysis and monitoring of the network port scanning process.

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] Ballmann, B., Understanding Network Hacks: Attack and Defense with Python 3 2nd ed. Springer, ISBN-10 3662621592, ISBN-13 978-3662621592, 2022, p. 236.
[4] Ballmann, B., Network Hacks - Intensivkurs: Angriff und Verteidigung mit Python 3 (German Edition). Springer, ISBN-10 3662616351, ISBN-13 2020, 978-3662616352, p. 237.
[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] Choi, B., Introduction to Python Network Automation: The First Journey 1st ed. Edition. Apress, ISBN-10 1484268059, ISBN-13 978-1484268056, 2021, p. 896.
[10] 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.
[11] 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.

[12] 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.
[13] 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.
[14] 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.
[15] Genov, B., Nedelchev, D., Mihovski, M., Mirchev, Y., Comprehensive approach for service life assessment of solid-propellant rocket motors. International Journal "NDT Days", Volume II, Issue 4, 2019, ISSN: 2603-4018 (print), 2603-4646 (online), pp. 467-475.
[16] Genov, B., NDT Assessment Model for Missile Motors. International Journal "NDT Days", Volume I, Issue 4, 2018, ISSN: 2603-4018 (print), 2603-4646 (online), pp. 484-493.
[17] Genov, B., Criteria for Selection of NDT in the Ammunition Life Cycle. International Journal "NDT Days", Volume I, Issue 4, 2018, ISSN: 2603-4018 (print), 2603-4646 (online), pp. 494-503.
[18] Genov, B., Kirkov, D., Mihovski, M., Mirchev, Y., Ageing of Solid Rocket Propellants Investigated by Ultrasound Technique. International Journal "NDT Days", Volume I, Issue 5, 2018, ISSN: 2603-4018 (print), 2603-4646 (online), pp. 577-582.
[19] Iliev, R., K. Ignatova. Implementation of cloud technologies for building data centers in defence and security. Information & Security: An International Journal 43, No. 1. 2019, ISSN 0861-5160, pp. 89-97., https://doi.org/10.11610/isij.4308.
[20] 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.

[21] Pavlova. D., Gindev, P. Designing an intelligent system for knowledge and process management in a university information environment. INTED2020 Proceedings. 14th International Technology, Education and Development Conference, Valencia, Spain, 2nd-4th March 2020, pp. 2743-2747. ISBN: 978-84-09-17939-8 doi: 10.21125/inted.2020.0818.

[22] 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.

[23] Pavlova, D., Gindev, P., System synthesis approach for intelligent knowledge 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. 250-257. ISSN 2367-7988.

[24] Pavlova, D., Gindev, P., A System for Intelligent Electronic Management of Knowledge and Business Processes in a University Information Environment. Proceedings of Seventh National Seminar with international participation - Intellectual property and digital people, 24 April 2019, Sofia, "ZA BUKVITE – O Pismeneh" Publishing House, vol. 7, 2019, pp. 221-234. ISBN 978-619-185-379-3.

[25] Radoeva, N., Iliev, R., A measurement process model implemented by generalized net. IEEE 8th International Conference on Intelligent Systems (IS), September 3-6, 2016, pp. 574-578, ISBN:978-1-5090-1355-5, DOI: 10.1109/IS.2016.7737482.