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

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

Original Contribution

DISCOVERING WIRELESS NETWORKS VIA MODIFIED SCRIPTS IMPLEMENTING LINEAR ALGORITHMS FOR 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 scripts, implementing linear algorithms for discovering wireless networks in Linux based operating systems are presented.

KEY WORDS: *Analysis, Bitrates, Channel, Connection, Encryption, Frequency, Linux, Modified scripts, Monitoring, Ports, Python3, Quality, Scanning, SSID, Traffic.*

1. Introduction

Wireless networks work mainly in two modes Ad-Hoc and Infrastructure. In the first mode of operation, it is typical for hosts to connect to each other. And in the second mode, the hosts connect to a central running device such as access point [2,3,4,5,6,7,9,14,15,18,19]. In practice, several basic modes are used to scan wireless networks. By default, network cards are set to normal scan mode. If the host wishes to scan all passing network frames from the wireless networks, its network card needs to be set in Promisc (monitor mode) [1,11,13,16,17,21,22,25,30,31,33]. All wireless networks work in infrastructure mode and in this regard every wireless network device sends special beacon network frames which broadcast SSID information to all nearby wireless devices. To find out about the operation of a particular wireless network, it is necessary for the wireless network card of the host to send a probe request. If the wireless network is available and working, it sends a probe response to the one who sent the probe request. The next logical step for the host is to establish a wireless network connection to the found wireless network, and to this end a series of authentication packets. Then it goes to the next stage with sending and receiving association request and response wireless network packets. The last procedure is related to performing a handshake via the following encryption security protocols - WPA, WPA2 and WPA3 [8,12,20,23,24,26,27,28,29,32]. After all phases of wireless network connection are passed, the corresponding host is connected to the found wireless network. The conducting wireless network scans that aim to reveal sensitive information about the wireless networks 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 scientific paper is only for research work and educational purposes and the author is not responsible in cases of abuse.

2. Experiment

The scientific experiment and research work in this paper in a controlled home environment was conducted. In this paper linear algorithms for discovering wireless networks via modified scripts for Linux based operating systems are suggested. In this regard, fundamentally new approaches for algorithmization of activities related to discovering of wireless networks are developed.

The programming language used for the scripts operating basically in Linux based operating systems is python version 3. The operation of the first modified script implementing a linear algorithm for discovering wireless networks involves the following basic steps [2,3,4,5,6,7,8,9,10,16,14,19]:

1. Specifying the full path to Python version 3.
2. Loading required modules and libraries – threating, datetime, socket, struct, time, ctypes, sys, sniff, os, scapy, wifi, columnar, IPy, argparse, colorama, Fore, queue and termcolor.
3. Defining the colors used in the Wi-Fi scanner.
4. Color script configuration.
5. Entering a network interface to scan – “wlan0”.
6. Discovering the available networks - SSIDs.
7. Showing the information about the wireless network with the best signal and parameters – channel, signal, frequency and encryption mode.
8. Displaying the elapsed time of the scan in seconds.
9. Showing the completion scan time.

The flowchart of the first modified script, implementing a linear algorithm for discovering wireless networks in Linux based operating systems on fig. 1 is illustrated.

The second script is characterized in that it detects all wireless probe requests and probe responses as well as service set identifiers (SSIDs) of wireless networks between hosts. It involves the following basic steps:

1. Specifying the full path to Python version 3.

2. Loading required modules and libraries – threating, datetime, socket, struct, time, ctypes, sys, sniff, os, scapy, wifi, columnar, IPy, argparse, colorama, Fore, queue and termcolor.

3. Putting the wireless card in monitoring mode.

4. Selecting the wireless interface – “wlan0”.

5. Printing the discovered SSIDs.

The flowchart of the second modified script, implementing a linear algorithm for discovering probe requests and probe responses in Linux based operating systems on fig. 2 is presented.

It should be specified that the script intercepts absolutely every wireless request and response between the hosts.

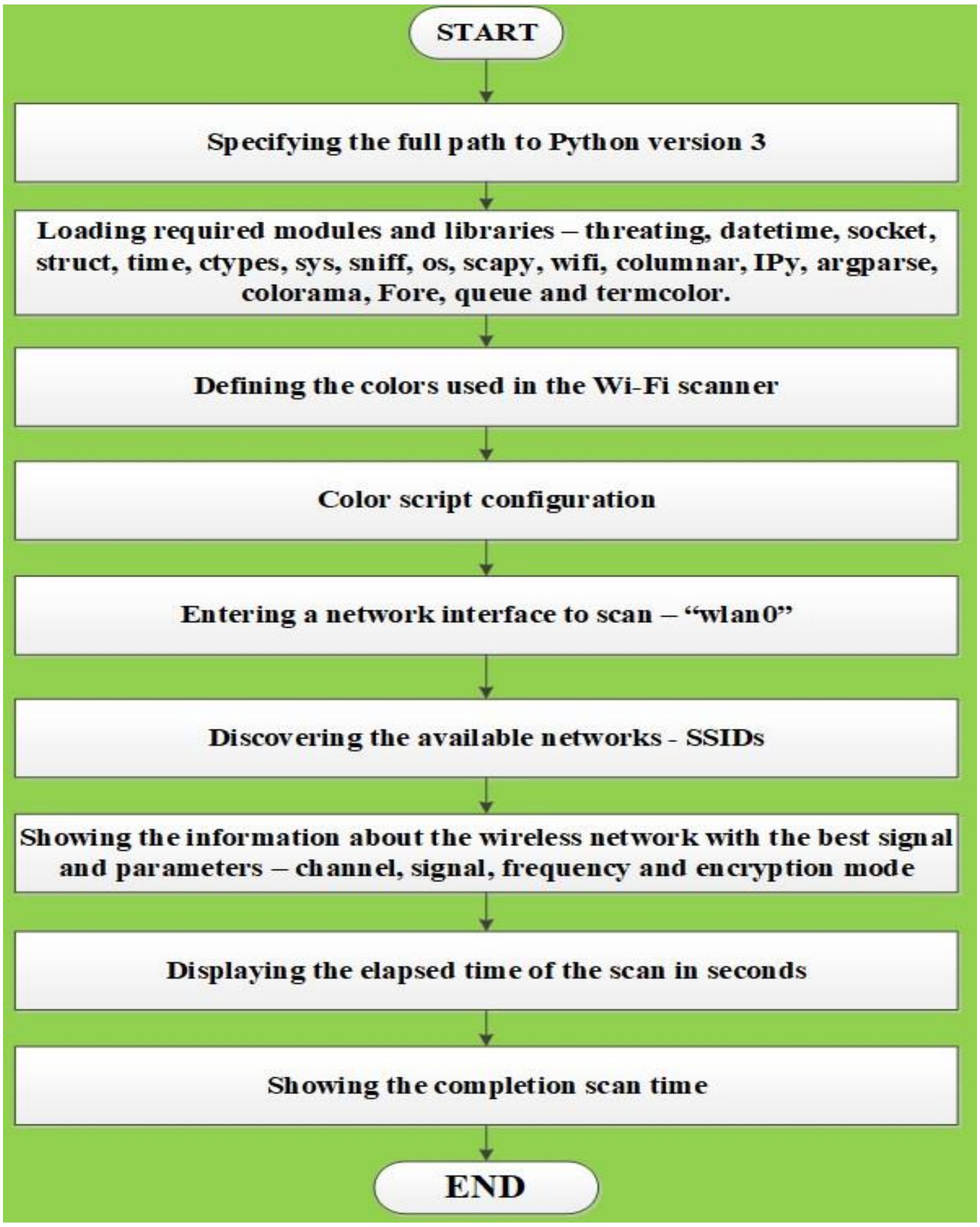

START
Specifying the full path to Python version 3
Loading required modules and libraries – threating, datetime, socket, struct, time, ctypes, sys, sniff, os, scapy, wifi, columnar, IPy, argparse, colorama, Fore, queue and termcolor.
Defining the colors used in the Wi-Fi scanner
Color script configuration
Entering a network interface to scan – “wlan0”
Discovering the available networks - SSIDs
Showing the information about the wireless network with the best signal and parameters – channel, signal, frequency and encryption mode
Displaying the elapsed time of the scan in seconds
Showing the completion scan time
END

Fig. 1. Flowchart of the first modified script, implementing a linear algorithm for discovering wireless networks

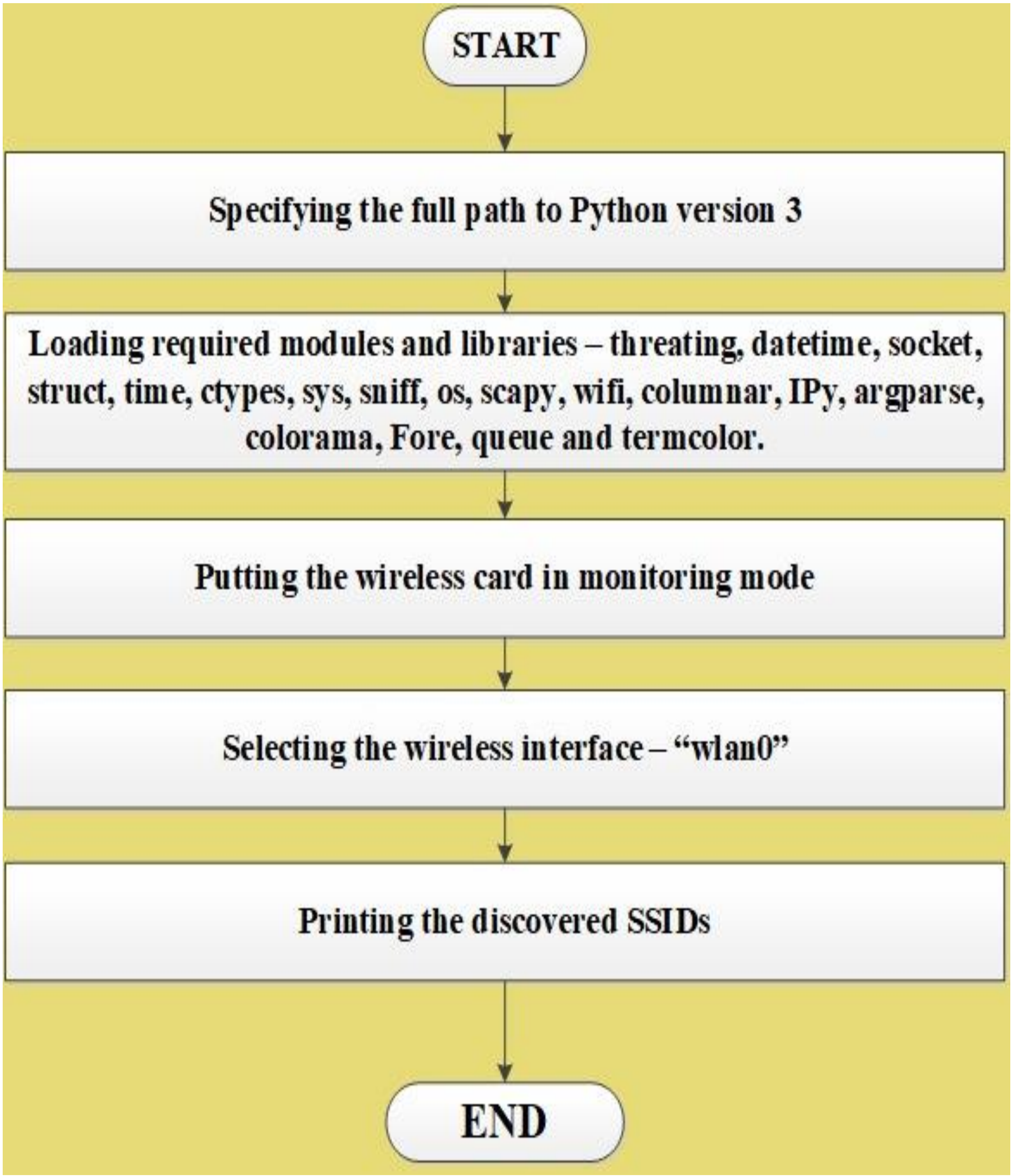

START
Specifying the full path to Python version 3
Loading required modules and libraries – threating, datetime, socket, struct, time, ctypes, sys, sniff, os, scapy, wifi, columnar, IPy, argparse, colorama, Fore, queue and termcolor.
Putting the wireless card in monitoring mode
Selecting the wireless interface – "wlan0"
Printing the discovered SSIDs
END

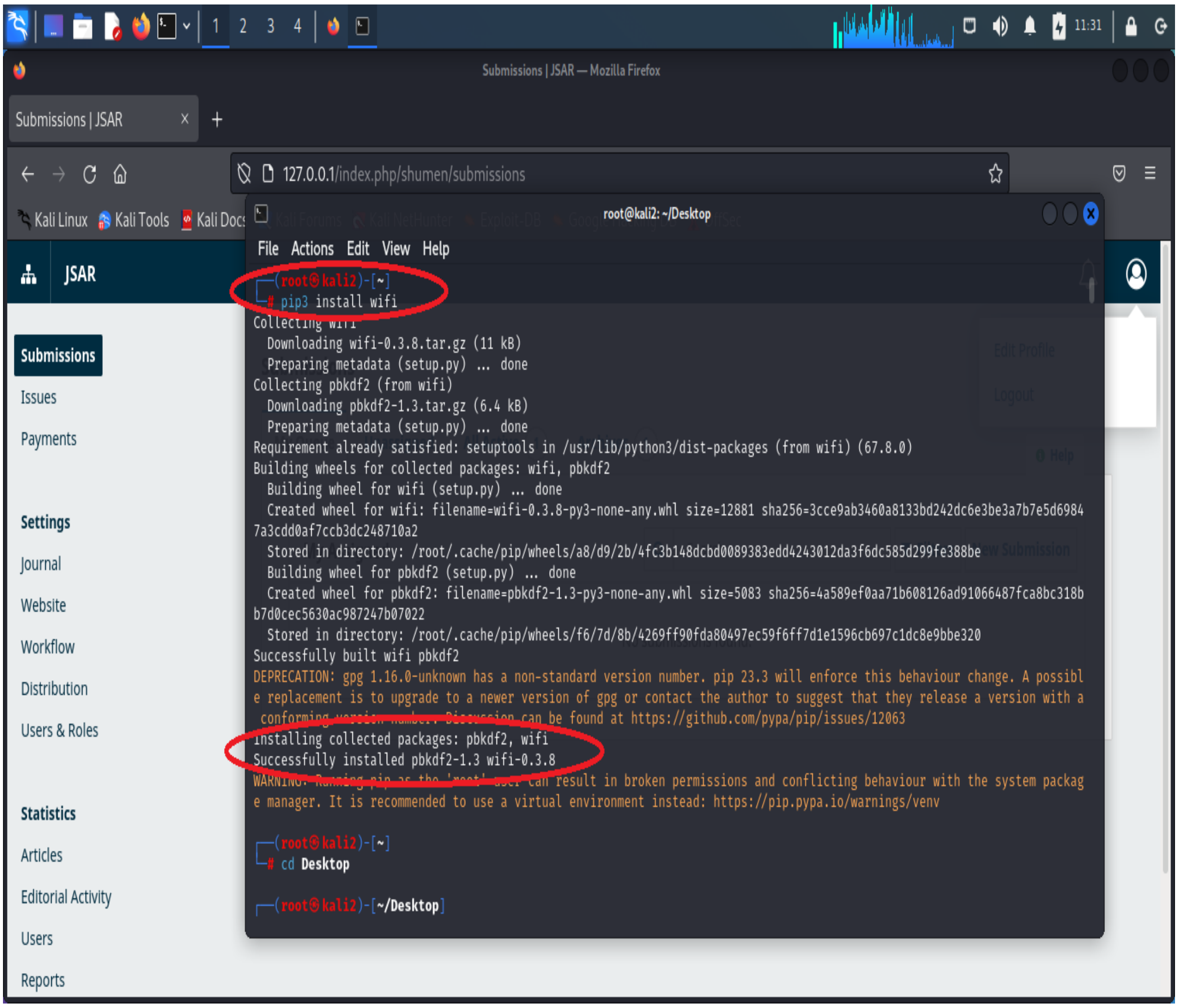
Fig. 2. Flowchart of the second modified script, implementing a linear algorithm for discovering probe requests and probe responses in Linux based operating systems

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 be discovered all wireless networks. The virtual installed operating system for the 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 probe requests and probe responses 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.

It was purchased additional TP-LINK TL-WN821N 300Mbps wireless N USB adapter in order to be carried out the scientific research. The IPv4 address of the scanning host is 192.168.80.130/24 and thus the both modified scripts, implementing a linear algorithm for discovering wireless networks on the host 192.168.80.130 with Kali Linux virtual operating system are executed.

Another important step that was taken is the installation of the two modules – wifi and scapy. Thanks to them, it is possible to be scanned and detected the wireless networks and putted the wireless network card in monitoring mode. This is shown on fig. 3.

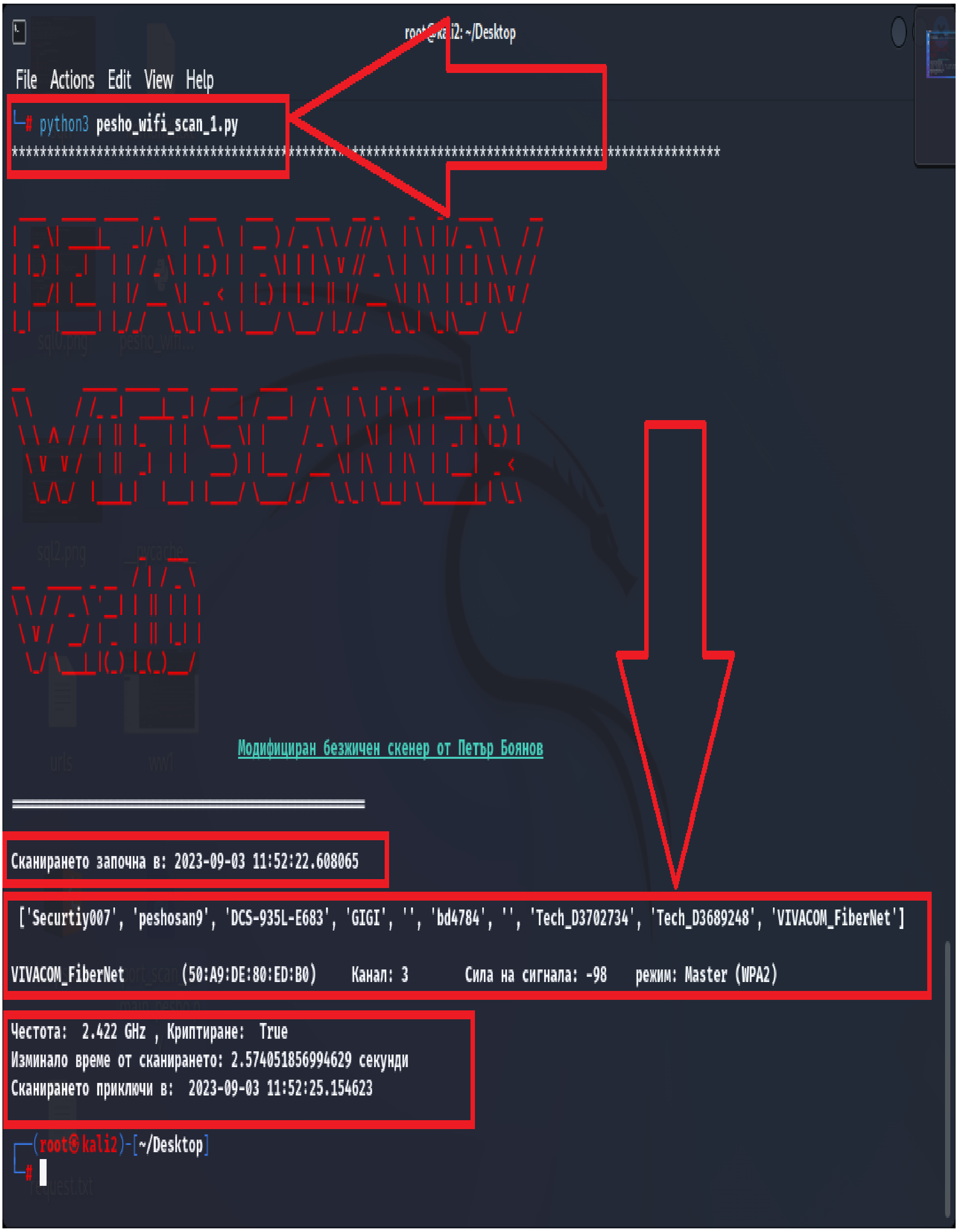


```
┌──(root㉿kali2)-[~]
└─# pip3 install wifi
Collecting wifi
  Downloading wifi-0.3.8.tar.gz (11 kB)
  Preparing metadata (setup.py) ... done
Collecting pbkdf2 (from wifi)
  Downloading pbkdf2-1.3.tar.gz (6.4 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: setuptools in /usr/lib/python3/dist-packages (from wifi) (67.8.0)
Building wheels for collected packages: wifi, pbkdf2
  Building wheel for wifi (setup.py) ... done
  Created wheel for wifi: filename=wifi-0.3.8-py3-none-any.whl size=12881 sha256=3cce9ab3460a8133bd242dc6e3be3a7b7e5d6984
7a3cdd0af7ccb3dc248710a2
  Stored in directory: /root/.cache/pip/wheels/a8/d9/2b/4fc3b148dcbd0089383edd4243012da3f6dc585d299fe388be
  Building wheel for pbkdf2 (setup.py) ... done
  Created wheel for pbkdf2: filename=pbkdf2-1.3-py3-none-any.whl size=5083 sha256=4a589ef0aa71b608126ad91066487fca8bc318b
b7d0cec5630ac987247b07022
  Stored in directory: /root/.cache/pip/wheels/f6/7d/8b/4269ff90fda80497ec59f6ff7d1e1596cb697c1dc8e9bbe320
Successfully built wifi pbkdf2
DEPRECATION: gpg 1.16.0-unknown has a non-standard version number. pip 23.3 will enforce this behaviour change. A possibl
e replacement is to upgrade to a newer version of gpg or contact the author to suggest that they release a version with a
 conforming version number. Discussion can be found at https://github.com/pypa/pip/issues/12063
Installing collected packages: pbkdf2, wifi
Successfully installed pbkdf2-1.3 wifi-0.3.8
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system packag
e manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

┌──(root㉿kali2)-[~]
└─# cd Desktop

┌──(root㉿kali2)-[~/Desktop]
```

Fig. 3. Installation of the module wifi in the operating system Kali Linux

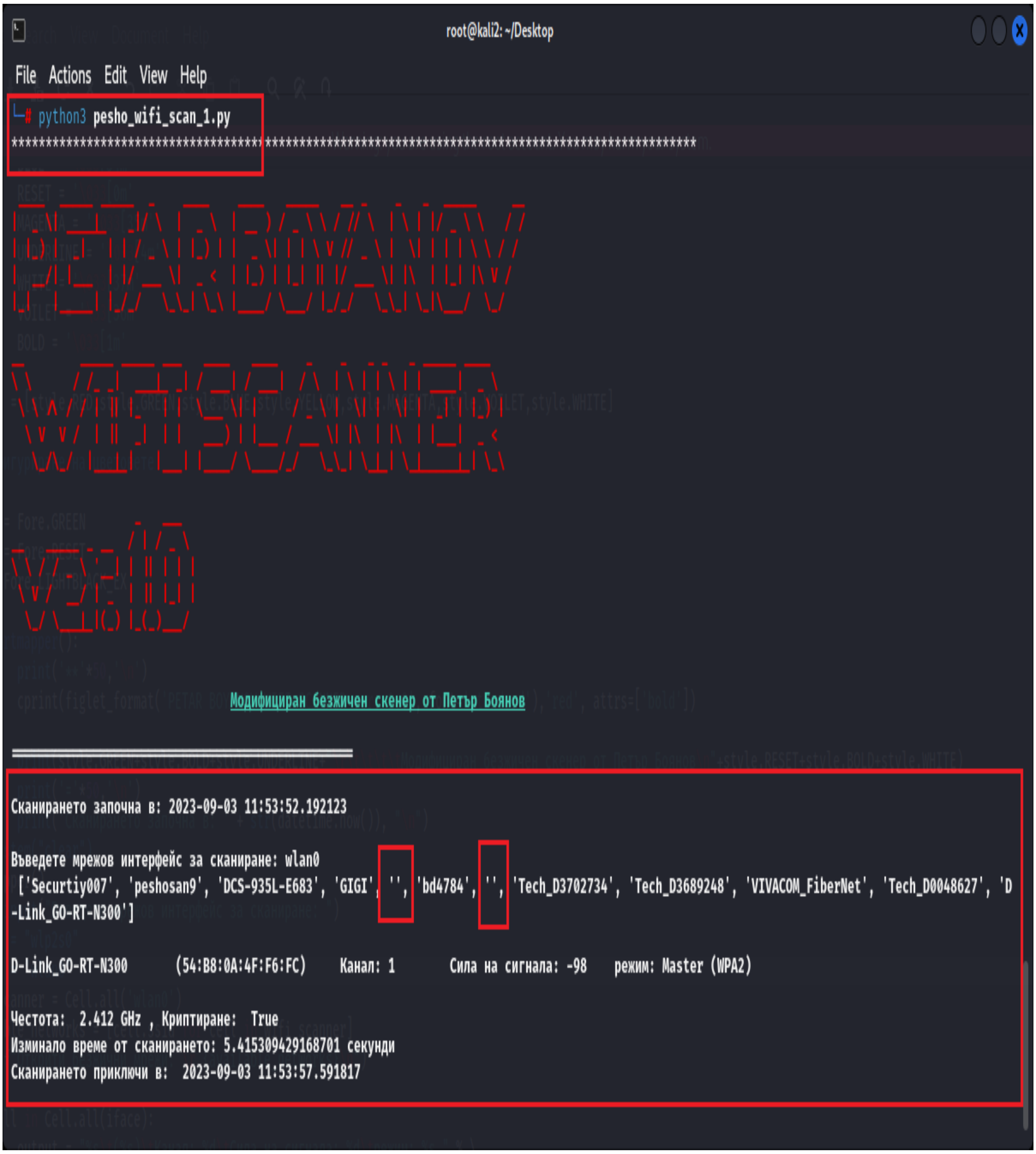
3. Results

Figures 4 and 5 show the results obtained after performing a wireless network scan with the first modified script on the host with address 192.168.80.130 using the command "python3 pesho_wifi_scan_1.py"



```
└─# python3 pesho_wifi_scan_1.py
********************************************************************************************

                         Модифициран безжичен скенер от Петър Боянов

═══════════════════════════════════════

Сканирането започна в: 2023-09-03 11:52:22.608065

 ['Securtiy007', 'peshosan9', 'DCS-935L-E683', 'GIGI', '', 'bd4784', '', 'Tech_D3702734', 'Tech_D3689248', 'VIVACOM_FiberNet']

VIVACOM_FiberNet     (50:A9:DE:80:ED:B0)     Канал: 3       Сила на сигнала: -98    режим: Master (WPA2)

Честота:  2.422 GHz , Криптиране:  True
Изминало време от сканирането: 2.574051856994629 секунди
Сканирането приключи в:  2023-09-03 11:52:25.154623

┌──(root㉿kali2)-[~/Desktop]
└─# 
```

Fig. 4. Discovered all wireless networks and presented the current SSID with the best indicators



```
└─# python3 pesho_wifi_scan_1.py
```

```
Модифициран безжичен скенер от Петър Боянов

Сканирането започна в: 2023-09-03 11:53:52.192123

Въведете мрежов интерфейс за сканиране: wlan0
 ['Securtiy007', 'peshosan9', 'DCS-935L-E683', 'GIGI', '', 'bd4784', '', 'Tech_D3702734', 'Tech_D3689248', 'VIVACOM_FiberNet', 'Tech_D0048627', 'D
-Link_GO-RT-N300']

D-Link_GO-RT-N300      (54:B8:0A:4F:F6:FC)     Канал: 1       Сила на сигнала: -98    режим: Master (WPA2)

Честота:  2.412 GHz , Криптиране:  True
Изминало време от сканирането: 5.415309429168701 секунди
Сканирането приключи в:  2023-09-03 11:53:57.591817
```

Fig. 5. Discovered all wireless networks and presented the current SSID with the best indicators

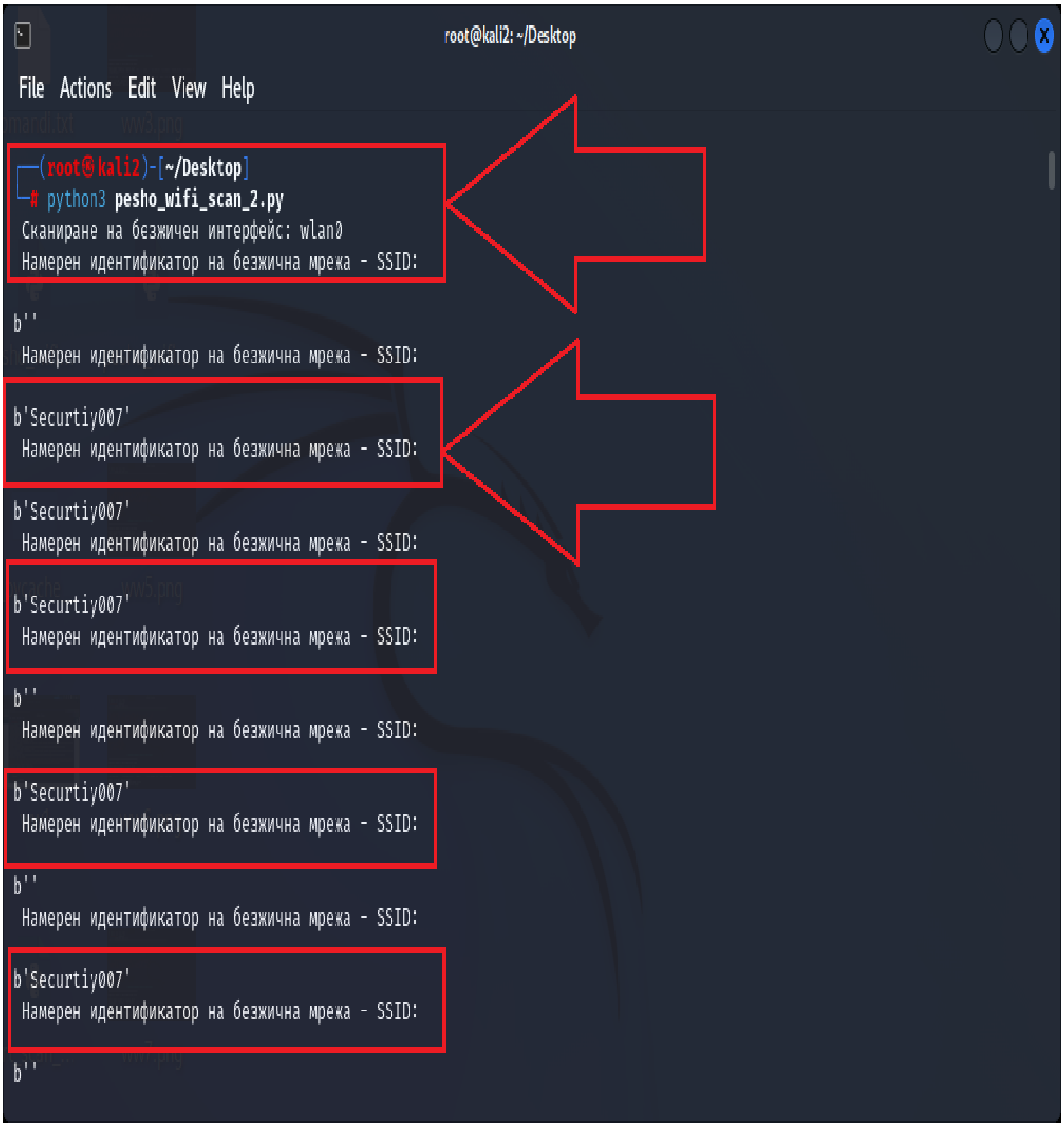
The obtained results from fig. 4 and 5 present that the following wireless networks were discovered:

- Security007;
- peshosan9;
- DCS-935-E683;
- GIGI;
- Two hidden wireless networks;
- bd4784;

- Tech_D3702734;
- Tech_D3689248;
- VIVACOM_FiberNet;
- Tech_D0048627;
- D-Link_Go-RT-N300.

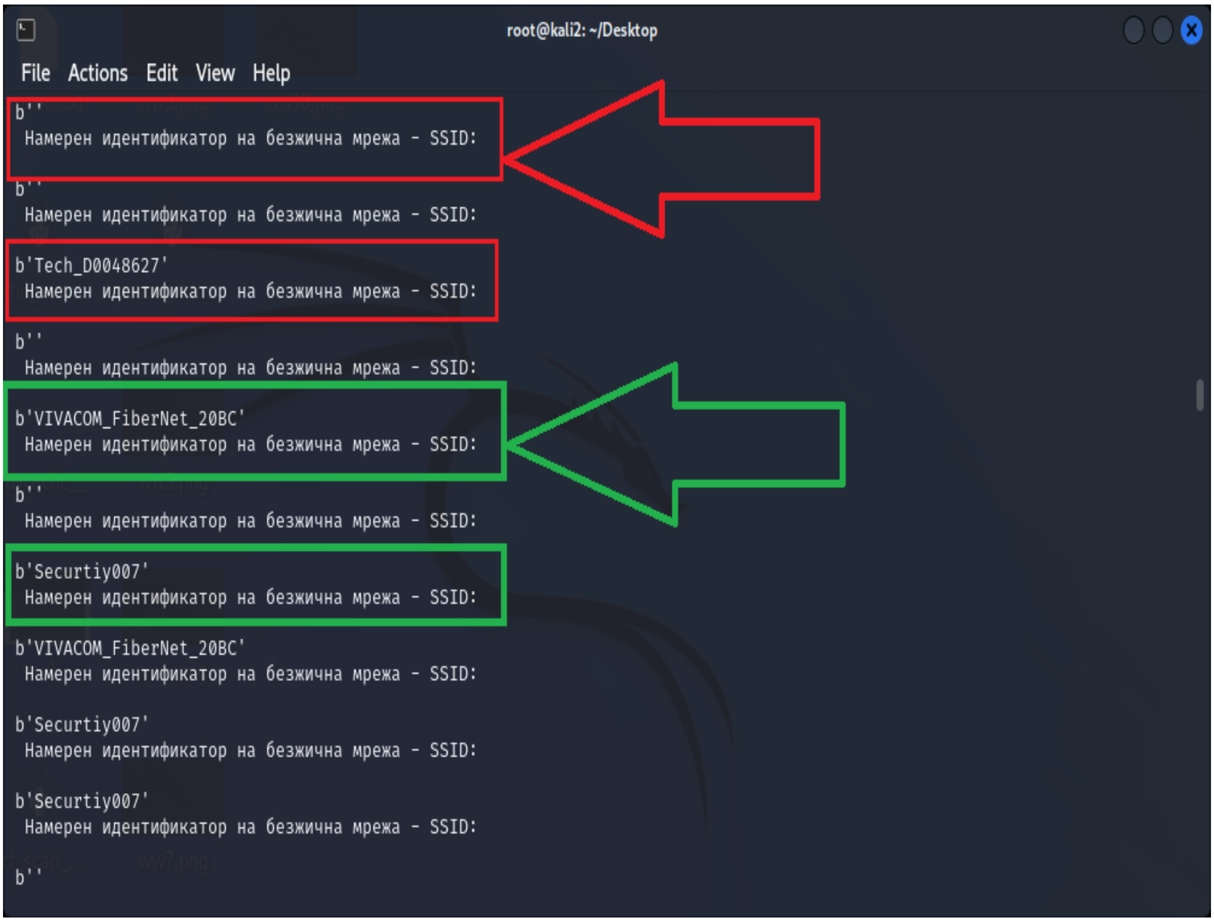
The discovered SSIDs “VIVACOM_FiberNet” and “D-Link_Go-RT-N300” has the best indicators. For the record my wireless networks are “peshosan9”, “VIVACOM_FiberNet_20BC”, “VIVACOM_FiberNet”, “Security007” and “DCS-935-E683”.

Figures 6, 7 and 8 show the results obtained after performing a wireless network scan with the second modified script on the host with address 192.168.80.130 using the command "python3 pesho_wifi_scan_2.py"



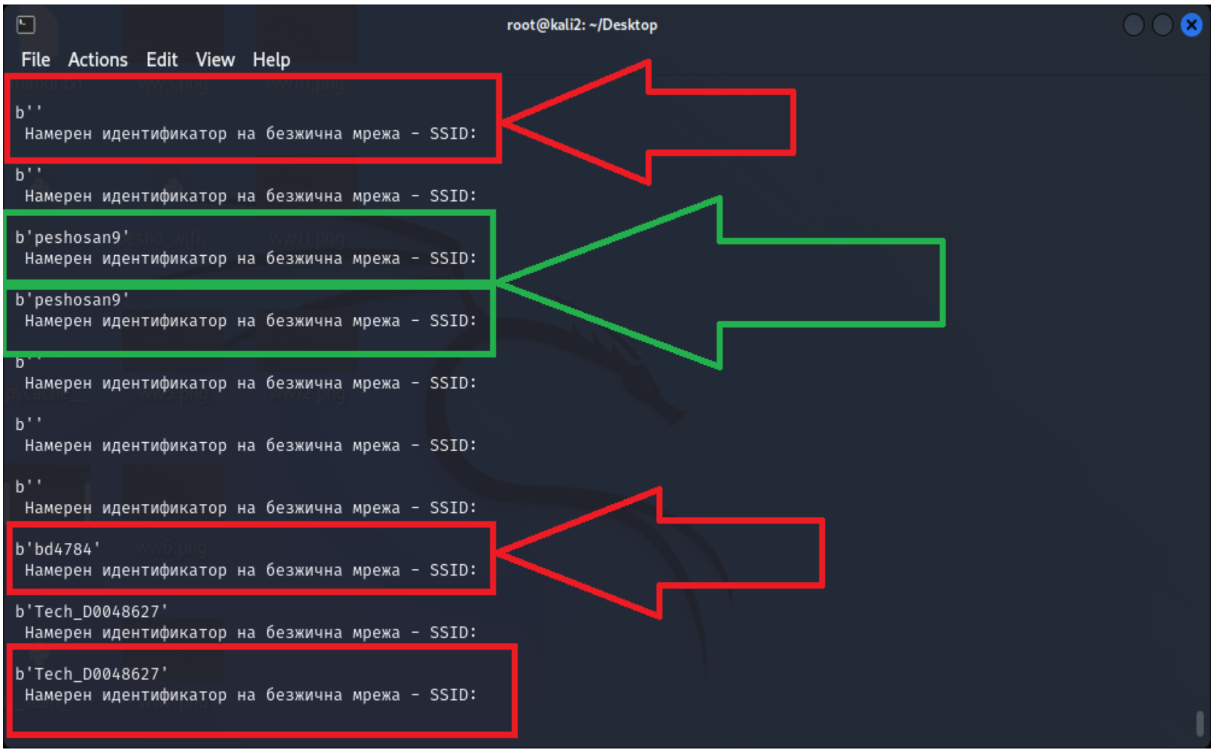
```
┌──(root㉿kali2)-[~/Desktop]
└─# python3 pesho_wifi_scan_2.py
Сканиране на безжичен интерфейс: wlan0
Намерен идентификатор на безжична мрежа - SSID:
b''
Намерен идентификатор на безжична мрежа - SSID:
b'Securtiy007'
Намерен идентификатор на безжична мрежа - SSID:
b'Securtiy007'
Намерен идентификатор на безжична мрежа - SSID:
b'Securtiy007'
Намерен идентификатор на безжична мрежа - SSID:
b''
Намерен идентификатор на безжична мрежа - SSID:
b'Securtiy007'
Намерен идентификатор на безжична мрежа - SSID:
b''
Намерен идентификатор на безжична мрежа - SSID:
b'Securtiy007'
Намерен идентификатор на безжична мрежа - SSID:
b''
```

Fig. 6. Discovered all sent probe requests and probe responses



```
b''
 Намерен идентификатор на безжична мрежа - SSID:

b''
 Намерен идентификатор на безжична мрежа - SSID:

b'Tech_D0048627'
 Намерен идентификатор на безжична мрежа - SSID:

b''
 Намерен идентификатор на безжична мрежа - SSID:

b'VIVACOM_FiberNet_20BC'
 Намерен идентификатор на безжична мрежа - SSID:

b''
 Намерен идентификатор на безжична мрежа - SSID:

b'Securtiy007'
 Намерен идентификатор на безжична мрежа - SSID:

b'VIVACOM_FiberNet_20BC'
 Намерен идентификатор на безжична мрежа - SSID:

b'Securtiy007'
 Намерен идентификатор на безжична мрежа - SSID:

b'Securtiy007'
 Намерен идентификатор на безжична мрежа - SSID:

b''
```

Fig. 7. Discovered all sent probe requests and probe responses



```
b''
 Намерен идентификатор на безжична мрежа - SSID:
b''
 Намерен идентификатор на безжична мрежа - SSID:
b'peshosan9'
 Намерен идентификатор на безжична мрежа - SSID:
b'peshosan9'
 Намерен идентификатор на безжична мрежа - SSID:
b''
 Намерен идентификатор на безжична мрежа - SSID:
b''
 Намерен идентификатор на безжична мрежа - SSID:
b''
 Намерен идентификатор на безжична мрежа - SSID:
b'bd4784'
 Намерен идентификатор на безжична мрежа - SSID:
b'Tech_D0048627'
 Намерен идентификатор на безжична мрежа - SSID:
b'Tech_D0048627'
 Намерен идентификатор на безжична мрежа - SSID:
```

Fig. 8. Discovered all sent probe requests and probe responses

The results of the conducted scientific research show that the modified scripts are able to discover all wireless networks and illustrate the process of sending probe requests and receiving probe responses. The discovering time of each wireless network scan is on the order of a few seconds (2.57 and 5.42 seconds).

3. Conclusion

The results obtained show that the two modified scripts can detect all wireless networks according to the technical capabilities of the TP-LINK TL-WN821N 300Mbps wireless N USB adapter. Thanks to these two developed algorithms, students studying in a professional area 5.3. Communication and computer networks will be able to learn to program the network socket and configure various settings and parameters when detecting wireless networks. 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 wireless networks discovering as well as intercepting wireless frame probe requests and probe responses.

Acknowledgments

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

References:

[1] Abedi, N., Bhaskar, A., Chung, E. and Miska, M., 2015. Assessment of antenna characteristic effects on pedestrian and cyclists travel-time estimation based on Bluetooth and WiFi MAC addresses. Transportation Research Part C: Emerging Technologies, 60, pp.124-141.

[2] Abbott-Jard, M., Shah, H. and Bhaskar, A., 2013, October. Empirical evaluation of Bluetooth and Wifi scanning for road transport. In Australasian Transport Research Forum (ATRF), 36th (Vol. 14).

[3] 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.

[4] 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.

[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] Čavojský, M., Uhlar, M., Ivanis, M., Molnar, M. and Drozda, M., 2018, August. User trajectory extraction based on wifi scanning. In 2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW) (pp. 115-120). IEEE.

[7] Castignani, G., Montavont, N. and Arcia-Moret, A., 2010. Analysis and evaluation of wifi scanning strategies. Proceeding of IV Cibelec, pp.3-7.

[8] Chilipirea, C., Petre, A.C., Dobre, C. and van Steen, M., 2016, June. Presumably simple: monitoring crowds using WiFi. In 2016 17th IEEE International Conference on Mobile Data Management (MDM) (Vol. 1, pp. 220-225). IEEE.

[9] Choi, T., Chon, Y. and Cha, H., 2017. Energy-efficient WiFi scanning for localization. Pervasive and Mobile Computing, 37, pp.124-138.

[10] Denis, M., Zena, C. and Hayajneh, T., 2016, April. Penetration testing: Concepts, attack methods, and defense strategies. In 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-6). IEEE.

[11] Fulara, H., Singh, G., Jaisinghani, D., Maity, M., Chakraborty, T. and Naik, V., 2019, June. Use of machine learning to detect causes of unnecessary active scanning in WiFi networks. In 2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM) (pp. 1-9). IEEE.

[12] Hidayat, A., Terabe, S. and Yaginuma, H., 2018. WiFi scanner technologies for obtaining travel data about circulator bus passengers: Case study in Obuse, Nagano prefecture, Japan. Transportation Research Record, 2672(45), pp.45-54.

[13] Hidayat, A., Terabe, S. and Yaginuma, H., 2018. Determine non-passenger data from WiFi scanner data (MAC address), a case study: Romango bus, Obuse, Nagano prefecture, Japan. International Review for Spatial Planning and Sustainable Development, 6(3), pp.154-167.

[14] Hou, P., Deng, R., Guo, J., Chen, W., Li, X. and Yu, H.Z., 2021. A WiFi scanner in conjunction with disposable multiplex paper assay for the quantitation of disease markers in blood plasma. Analytical and Bioanalytical Chemistry, 413, pp.4625-4634.

[15] Hu, X., Song, L., Van Bruggen, D. and Striegel, A., 2015, October. Is there WiFi yet? How aggressive probe requests deteriorate energy and throughput. In Proceedings of the 2015 Internet Measurement Conference (pp. 317-323).

[16] Huang, Z., Xu, L. and Lin, Y., 2020. Multi-stage pedestrian positioning using filtered WiFi scanner data in an urban road environment. Sensors, 20(11), p.3259.

[17] 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.

[18] Liu, H., Yang, J., Sidhom, S., Wang, Y., Chen, Y. and Ye, F., 2013. Accurate WiFi based localization for smartphones using peer assistance. IEEE Transactions on Mobile Computing, 13(10), pp.2199-2214.

[19] Liu, H., Gan, Y., Yang, J., Sidhom, S., Wang, Y., Chen, Y. and Ye, F., 2012, August. Push the limit of WiFi based localization for smartphones. In Proceedings of the 18th annual international conference on Mobile computing and networking (pp. 305-316).

[20] Lu, H.J. and Yu, Y., 2021. Research on WiFi penetration testing with Kali Linux. Complexity, 2021, pp.1-8.

[21] Mehta, G. and Gaur, M.S., 2014. WiFi AP Info. Exploitation and Packet Capturing on Android Devices using CONNECTOR & KALI LINUX.

[22] Patra, S.S., Muthurajan, B., Chilukuri, B.R. and Devi, L., 2019, January. Development and evaluation of a low-cost wifi media access control scanner as traffic sensor. In 2019 11th International Conference on Communication Systems & Networks (COMSNETS) (pp. 777-782). IEEE.

[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] Petre, A.C., Chilipirea, C., Baratchi, M., Dobre, C. and van Steen, M., 2017. WiFi tracking of pedestrian behavior. In Smart Sensors Networks (pp. 309-337). Academic Press.

[25] Sakib, M.N., Halim, J.B. and Huang, C.T., 2014, December. Determining location and movement pattern using anonymized WiFi access point BSSID. In 2014 7th International Conference on Security Technology (pp. 11-14). IEEE.

[26] Sapiezynski, P., Stopczynski, A., Wind, D.K., Leskovec, J. and Lehmann, S., 2017. Inferring person-to-person proximity using WiFi signals. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(2), pp.1-20.

[27] Sapiezynski, P., Stopczynski, A., Gatej, R. and Lehmann, S., 2015. Tracking human mobility using wifi signals. PloS one, 10(7), p.e0130824.

[28] Shi, J., Meng, L., Striegel, A., Qiao, C., Koutsonikolas, D. and Challen, G., 2016, April. A walk on the client side: Monitoring enterprise wifi networks using smartphone channel scans. In IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications (pp. 1-9). IEEE.

[29] Sofyan, M., Wardriving WiFi menggunakan Wigle pada Area Kambang Iwak Palembang. Wardriving WiFi menggunakan Wigle pada Area Kambang Iwak Palembang.

[30] Vu, L., Nahrstedt, K., Retika, S. and Gupta, I., 2010, October. Joint bluetooth/wifi scanning framework for characterizing and leveraging people movement in university campus. In Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems (pp. 257-265).

[31] Vu, L., 2010. Lessons learned from bluetooth/wifi scanning deployment in university campus.

[32] Wang, L., Chen, C.T. and Tsai, C.M., 2023, July. Research on cracking WIFI wireless network using Kali-Linux penetration testing software. In Third International Conference on Digital Signal and Computer Communications (DSCC 2023) (Vol. 12716, pp. 378-382). SPIE.

[33] Wang, W., Joshi, R., Kulkarni, A., Leong, W.K. and Leong, B., 2013, July. Feasibility study of mobile phone WiFi detection in aerial search and rescue operations. In Proceedings of the 4th Asia-Pacific workshop on systems (pp. 1-6).