Taking LOLBAS Hacking to Another Level—Stealing Passwords using Built-in Binaries*

Maksim Iavich[1,†], Sergiy Gnatyuk[2,*,†], Sergei Simonov[1,†] and Viktoriia Sydorenko[2,†]

[1] Caucasus University, 1 Paata Saakadze str., 0102 Tbilisi, Georgia
[2] National Aviation University, 1 Liubomyra Huzara ave., 03058 Kyiv, Ukraine

Abstract

This paper examines the use of Living Off The Land Binaries and Scripts (LOLBAS) as a method of exploiting Windows environments for the exfiltration of sensitive data, such as passwords and files. LOLBAS techniques involve leveraging legitimate, pre-existing system binaries and scripts to carry out malicious activities, making them harder to detect by traditional security measures. In this study, two custom binaries are demonstrated, utilizing native Windows tools to create covert data exfiltration channels and establish unauthorized administrative access. The paper provides an overview of the methods used to exploit these tools, discusses the outcomes and implications of the results, and offers suggestions for future research into more effective countermeasures to mitigate these types of attacks. The primary focus of this research is to evaluate how well-existing antivirus programs can detect and respond to relatively simple LOLBAS techniques, highlighting potential gaps in current detection capabilities.

Keywords

LOLBAS hacking, AV bypass, zero-day attacks, penetration testing

1. Introduction

With the continuous evolution of modern operating systems, an increasing number of built-in binaries and scripts have been introduced, significantly expanding the attack surface for Living Off The Land (LOTL) techniques. These native utilities are designed to enhance system functionality, improve user experience, and streamline system administration. However, with each new utility added, the potential for malicious misuse also grows. For example, recent Windows updates have incorporated new features such as OpenSSH, which is aimed at improving secure communications, but simultaneously opens a door for attackers to exploit the system. These built-in tools are inherently trusted by the operating system and often bypass traditional security measures, making them an ideal target for cybercriminals looking to exploit the system without raising any red flags. Because these tools are considered normal system components, it becomes increasingly difficult for traditional security tools, which primarily focus on identifying standalone malicious software, to distinguish between legitimate administrative actions and malicious activities. The Living Off The Land (LOTL) approach exploits the trusted nature of these built-in binaries and scripts by repurposing them for malicious purposes, thus allowing attackers to evade detection. By using these legitimate utilities, attackers can blend in with normal system operations, making it significantly harder to differentiate between normal and malicious use. As operating systems continue to evolve and incorporate even more native utilities and scripts, the potential avenues for exploitation increase, offering more opportunities for malicious actors to bypass security measures and evade detection. This study specifically examines practical methods by which attackers can leverage native Windows utilities for malicious purposes, focusing on two primary objectives: the

*CPITS 2025: Workshop on Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2025, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
miavich@cu.edu.ge (M. Iavich); simonovi@cu.edu.ge (S. Simonov); sergio.gnatyuk@gmail.com (S. Gnatyuk); v.sydorenko@ukr.net (V. Sydorenko)
0000-0002-3109-7971 (M. Iavich); 0009-0000-0124-2931 (S. Simonov); 0000-0003-4992-0564 (S. Gnatyuk); 0000-0002-5910-0837 (V. Sydorenko)

exfiltration of sensitive data and the establishment of persistent unauthorized access. These objectives are achieved without the need to introduce any external software or tools into the system. By relying solely on built-in system tools, attackers can reduce their footprint, maintain stealth, and increase their chances of evading detection by security software. This method not only reduces the risk of detection but also allows attackers to carry out their operations with minimal disruption to the target system, making it difficult for defenders to spot any abnormalities or signs of compromise.

To illustrate these techniques in practice, this study developed and tested two custom binaries that exploit native Windows tools for malicious purposes. These binaries demonstrate how attackers can manipulate trusted system tools to exfiltrate sensitive data and create a backdoor for persistence. Detailed examples of these implementations are provided, highlighting their potential impact on security and the challenges faced by defenders when attempting to detect and mitigate such attacks. The study aims to provide valuable insights into the risks associated with LOTL techniques, particularly focusing on how they enable attackers to bypass traditional security measures. By understanding these attack vectors, defenders can better prepare for and develop countermeasures against these types of threats. This research underscores the critical importance of closely monitoring the use of built-in binaries and scripts, particularly in environments where security is paramount. It also emphasizes the need for more advanced detection techniques that go beyond traditional methods and can identify unusual or suspicious behavior even when using trusted legitimate system tools.

The goal of this research is not only to analyze and demonstrate the effectiveness of LOLBAS hacking techniques in bypassing well-known antivirus software but also to provide a foundation for developing better detection and defense strategies. By exploring the vulnerabilities created by the abuse of built-in system utilities, this study aims to contribute to the development of more effective security measures that can counter the increasing sophistication of LOTL-based attacks. This research will help inform future security frameworks, guiding efforts to enhance system monitoring, improve threat detection, and reduce the risks associated with advanced cyberattack techniques.

2. Review of the literature

Modifying the template—including but not limited to: adjusting margins, typeface sizes, line spacing, paragraph and list definitions—is not allowed. Workshop organizers may want to provide a copy of this template to authors where the event title in the footnote is updated to their workshop details, see “Woodstock …” footnote on page 1. While we provide a Word/LibreOffice template, we strongly recommend authors use our LaTeX template.

The literature on techniques and detection methods for Living-Off-The-Land Binary (LOLBin) and scripting (LOLBas) attacks offers a broad spectrum of insights into this area of cybersecurity. The following review highlights key contributions to this domain. In [1], the authors examine the technical capabilities of SSH port forwarding in their foundational work. While not directly addressing LOLBin or LOLBas techniques, their study provides a basis for understanding how secure protocols can be used in unconventional ways, including potentially malicious purposes. Research [2] contributes by analyzing the limitations of antivirus (AV) dynamic analysis models and detailing methods for bypassing AV detection. The paper [3] focuses on malware written in low-level languages such as C and Rust. By comparing the characteristics of malware created in these languages, the study provides useful context for understanding how programming language choices can influence the design and detectability of malicious code. The papers [4–6] propose ways of LOLBin attack detection using artificial intelligence. Research [7] extends the discussion by focusing on log correlation techniques for detecting attacks in Active Directory environments. Their findings stress the importance of integrating log data and correlation methods to identify threats effectively, including those based on LOLBin tactics. The Research [8] introduces active learning techniques for detecting living-off-the-land commands. The study demonstrates the

potential of machine learning to identify these subtle and often deceptive attack methods, highlighting the importance of training detection systems with diverse datasets. The paper [9] provides a systematic analysis of Windows-based malware that relies on living-off-the-land strategies. Their study identifies the underlying patterns and tactics used by attackers, offering critical insights into how these threats evolve and operate. The research [10] proposes a deep learning-based approach for detecting living-off-the-land attacks, emphasizing the scalability and adaptability of deep learning models in identifying such threats. This research underscores the value of leveraging modern AI techniques for advanced threat detection. In [7], the authors extend the discussion by focusing on log correlation techniques for detecting attacks in Active Directory environments. Their findings stress the importance of integrating log data and correlation methods to identify threats effectively, including those based on LOLBin tactics. By leveraging log correlation, the authors illustrate how detailed activity tracking within Active Directory environments can provide valuable insights for detecting lateral movement and privilege escalation. The paper [11] focuses on Endpoint Detection and Response (EDR) strategies for combating fileless malware and LOLBin threats. Their work highlights the practical application of EDR tools in real-world scenarios, showcasing their effectiveness in mitigating these attacks [12–14].

Collectively, these studies form a comprehensive foundation for understanding and addressing LOLBin and LOLBas threats. They range from foundational works on related techniques to advanced detection mechanisms using machine learning, pattern recognition, and deep learning. This body of literature provides a valuable resource for further research and practical application in defending against these evolving threats [15].

3. Methodology

The offered binary files were tested in the following environment:

- Operating System: Windows 10 Pro (Version 21H2).
- Programming languages: C.
- Utilities/Tools used: PowerShell, SSH/SCP, icacls, net.exe, reg.exe.
- Network Setup: Isolated virtual environment with a controlled SSH server. VirtualBox was used as a hypervisor.
- Defensive Tools: Windows Defender, Bitdefender.

The binary files offered by the paper have the following roles. File 1 (Stealer)—Downloads a private SSH key from a remote server. Configures permissions on the key to restrict access. Establishes an SSH connection to create directories remotely. Copies files from the local system to the remote server. File 2 (Backdoor)—Creates a new local user with administrative privileges. Modifies the registry to disable RemoteUAC. Downloads and configures an SSH key. Sets up a reverse SSH tunnel to enable remote access.

Below, you can observe the pseudo-code of the offered lolbas binaries:

Code 1.

1. Define a function `run_command(command)`:

    a. Hide the console window using `ShowWindow(hWnd, SW_HIDE)`.
    b. Execute the given command using the system function.

2. In the main program:

   a. Define the following commands as strings:
   - Command1: Download a file from a specified URL using PowerShell and save it to a public folder.
   - Command2: Remove inheritance permissions for the downloaded file using `icacls`.
   - Command3: Grant full permissions for the downloaded file to the current user using `icacls`.
   - Command4: Use SSH to create a directory on a remote server. The directory name is based on the computer's name.
   - Command5: Use SCP to copy a specific local folder (including its contents) to the remote server's directory.

   b. Call `run_command()` for each of the commands in the defined order:
   - Execute Command1.
   - Execute Command2.
   - Execute Command3.
   - Execute Command4.
   - Execute Command5.

3. End the program.

Code 2.

1. Define a function `run_command(command)`:

   a. Execute the given command using the system function.

2. In the main program:

   a. Get a handle to the console window using `GetConsoleWindow()`.
   b. Hide the console window using `ShowWindow(hWnd, SW_HIDE)`.
   c. Define the following commands as strings:
   - Command1: Create a new user account named "eve" with a password ("qwerty") using the `net user` command.
   - Command2: Add the "eve" user to the Administrators group using the `net localgroup` command.
   - Command3: Modify the Windows registry to allow remote administrative tasks with the `reg add` command.
   - Command4: Download a file from a specified URL using PowerShell and save it to the public folder.
   - Command5: Remove inheritance permissions for the downloaded file using `icacls`.
   - Command6: Grant full permissions for the downloaded file to the current user using `icacls`.
   - Command7: Establish a reverse SSH tunnel to a remote server using the `ssh` command and the downloaded key file.
   d. Call `run_command()` for each of the commands in the defined order.

3. End the program.

Both files are undetected by regular antivirus solutions and can be used to deal with harm in a production environment. Static analysis in being bypassed as the files do not contain malicious signatures. The dynamic analysis is being bypassed as the offered binaries only utilize operating system functions that are never considered malicious. The processes created upon the execution of the offered binary files are transparent and running in the background, which makes the attack harder to detect for regular users. As the traffic (data exfiltration or the backdoor utilization) between the victim and the server is being sent through the SSH, it is being encrypted, which, in addition, can bypass the network IDS solutions [16–19].

4. Results discussions

The research presented in this paper explores the development and effectiveness of LOLBAS binaries with a focus on bypassing traditional antivirus solutions. The primary goal of the research is to analyze the efficacy of popular antivirus programs in detecting simple LOLBAS hacking techniques. The first binary successfully downloaded an SSH key and adjusted its permissions, uploaded files from the local system to a remote server, and avoided detection by Windows Defender due to the usage of native binaries. The second binary successfully created a new administrative user, modified registry values to enable unrestricted remote access, established a persistent reverse SSH tunnel, and similarly evaded detection by leveraging trusted binaries. Both binaries remained undetected by standard antivirus solutions, emphasizing the stealth of LOLBAS techniques. Commands executed seamlessly without errors, leveraging the trust in built-in tools. However, the approach relies on initial access and the availability of certain utilities such as PowerShell and SSH, which means that the offered approach will not work on operating systems older than Windows 10. The demonstrated methods highlight critical vulnerabilities in Windows environments. The abuse of trusted tools allows attackers to bypass many traditional defenses. Unauthorized administrative access was achieved with minimal logging, and sensitive files were transferred without raising alerts. Below you may observe the result of the backdoor (hackme.exe) file execution. Also, you may observe the result of VirusTotal checks performed against binary files [16, 20].

```
root@Web-Pentest:~# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      1662088/systemd-res
tcp        0      0 127.0.0.1:445           0.0.0.0:*               LISTEN      3531266/sshd: hacke
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2158782/sshd: /usr/
tcp6       0      0 :::443                  :::*                    LISTEN      3531180/apache2
tcp6       0      0 ::1:445                 :::*                    LISTEN      3531266/sshd: hacke
tcp6       0      0 :::80                   :::*                    LISTEN      3531180/apache2
tcp6       0      0 :::22                   :::*                    LISTEN      2158782/sshd: /usr/
udp        0      0 127.0.0.53:53           0.0.0.0:*                           1662088/systemd-res
```

Figure 1: After the execution of the backdoor file (hackme.exe), the attacker machine has port 445 open as the result of SSH port forwarding

```
root@Web-Pentest:~# crackmapexec smb 127.0.0.1 -u eve -p qwerty -x "whoami"
SMB         127.0.0.1       445    DESKTOP-987VUU7  [*] Windows 10.0 Build 22621 (name:DESKTOP-987VUU7) (domain:DESKTOP-
987VUU7) (signing:False) (SMBv1:False)
SMB         127.0.0.1       445    DESKTOP-987VUU7  [+] DESKTOP-987VUU7\eve:qwerty (Pwn3d!)
SMB         127.0.0.1       445    DESKTOP-987VUU7  [+] Executed command via atexec
SMB         127.0.0.1       445    DESKTOP-987VUU7  nt authority\system
root@Web-Pentest:~#
```

Figure 2: The successful command execution is achieved using the crackmapexec tool. This is possible as the RemoteUAC was disabled on the victim's machine. This action was also performed by the backdoor script

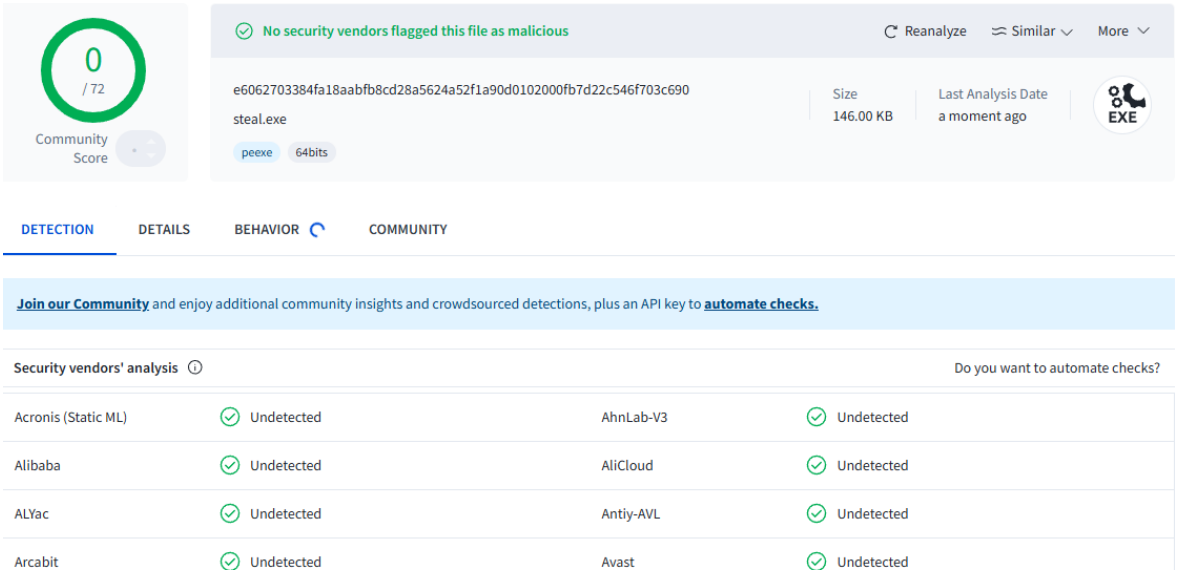

Figure 3: Results of the stealer (steal.exe) being checked by VirusTotal

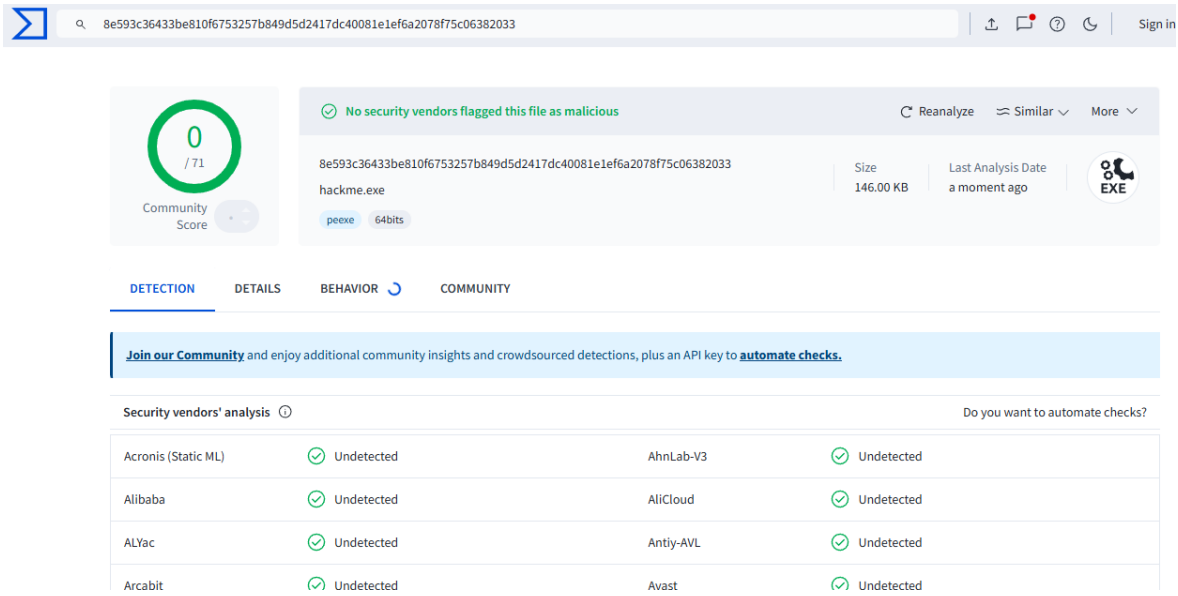

Figure 4: Results of the stealer (steal.exe) being checked by VirusTotal

This research explores the limitations of current antivirus solutions and highlights the importance of addressing ethical considerations. Tools used for penetration testing and security research should follow ethical guidelines to ensure responsible and lawful use. The information provided is intended to improve cybersecurity defenses and should not be used for harmful purposes.

Conclusions and future work

This paper demonstrates the effectiveness of LOLBAS in conducting stealthy attacks, focusing on data exfiltration and backdoor creation. By leveraging built-in binaries, attackers can evade traditional defenses and achieve their objectives with minimal traces.

Bypassing Antivirus Detection: The research demonstrates the effectiveness of using LOLBAS techniques to evade detection by antivirus software such as Windows Defender and Bitdefender. By leveraging trusted Windows utilities and avoiding the use of external malicious binaries, the developed tools remained undetected during both static and dynamic analysis.

Exploitation of Built-In Tools: The study highlights how native Windows utilities such as PowerShell, SSH, and registry modification commands can be abused for malicious purposes. These

tools, inherently trusted by the operating system, enable attackers to exfiltrate sensitive data and establish persistent unauthorized access with minimal footprint.

Minimal Logging and Stealth: The execution of the LOLBAS binaries produced limited logs, making detection challenging for traditional monitoring systems. The data exfiltration and backdoor creation processes leveraged encrypted communication channels, such as SSH, to further obfuscate malicious activities from network intrusion detection systems (IDS).

Impact of Windows Environment Features: The techniques employed relied on functionalities introduced in modern Windows environments, such as Windows 10. This highlights the dependence on specific utilities, which limits the applicability of these methods on older operating systems lacking similar tools.

Limitations of Traditional Defenses: The research underscores critical vulnerabilities in antivirus and IDS solutions, demonstrating their inability to effectively detect misuse of built-in binaries. This gap emphasizes the need for more robust detection mechanisms that focus on behavioral analysis rather than relying solely on signature-based approaches.

Future work in this field will involve the development of advanced detection algorithms and techniques tailored to identify the malicious use of LOLBAS tools, which often exploit trusted system resources for nefarious purposes. These algorithms would need to be more sophisticated than current antivirus solutions, which may struggle to detect these subtle, legitimate tools being repurposed for attacks. The research will focus on improving the precision and efficiency of these detection methods to ensure that they can reliably identify unusual or suspicious behaviors linked to the misuse of native Windows binaries and scripts. In addition to technological improvements, it is crucial to raise awareness within the cybersecurity community and among system administrators about the risks associated with granting unrestricted access to system binaries and scripts. By understanding the potential for these tools to be misused, organizations can take proactive steps to harden their environments. This includes reviewing and controlling access permissions to sensitive system components, ensuring that only trusted and necessary personnel can execute potentially dangerous binaries or scripts. Furthermore, the implementation of stricter security policies regarding script execution and binary usage is essential. These policies could include limiting the execution of unauthorized scripts, implementing strict whitelisting procedures for binaries, and monitoring the use of administrative privileges more closely. Organizations should also consider using application whitelisting to prevent unauthorized or suspicious binaries from running, thus mitigating the risk of exploitation. By enforcing these policies and integrating automated monitoring systems, companies can reduce the likelihood of successful LOLBAS-based attacks.

Finally, future research should explore the intersection of LOLBAS exploitation with other advanced cyberattack techniques, aiming to provide a more holistic approach to defending against this growing threat. By combining detection algorithms, user education, and stronger security controls, it will be possible to better mitigate the risks posed by LOLBAS exploitation, safeguarding critical infrastructure and sensitive data against increasingly sophisticated attackers.

Acknowledgments

This paper analyzed the following simultaneous localization and mapping (SLAM) algorithms: EKF SLAM, FastSLAM, Graph SLAM, SEIF SLAM, LIDAR SLAM, VSLAM, and IMU SLAM.

Combining algorithms can significantly improve the overall performance of SLAM systems. For exes.

Declaration on Generative AI

While preparing this work, the authors used the AI programs Grammarly Pro to correct text grammar and Strike Plagiarism to search for possible plagiarism. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the publication's content.

References

[1] G. Orr, J. Wyatt, {SSH} port forwarding, in: 4th Annual Linux Showcase & Conference (ALS 2000), 2000.

[2] E. Nasi, Bypass antivirus dynamic analysis, Limitations of the AV model and how to exploit them, 2014.

[3] M. K. Praveen, A comparative analysis of malware written in the C and Rust programming languages, Rochester Institute of Technology, 2023.

[4] T. Ongun, et al., Living-off-the-land command detection using active learning, in: 24th International Symposium on Research in Attacks, Intrusions and Defenses, 2021, 442–455.

[5] A. AbuShqeir, Common pattern generation for the detection of LOLBin attacks, Master's thesis, San Jose State University, 2023.

[6] U. Nisslmueller, LOLBin detection through unsupervised learning: An approach based on explicit featurization of the command line and parent-child relationships, Master's thesis, University of Twente, 2022.

[7] M. S. Elmastaş, C. Eyüpoğlu, Detection of current attacks in active directory environment with log correlation methods, J. Aeronautics Space Technol. 16(2) (2023) 36–55.

[8] T. Ongun, et al., Living-off-the-land command detection using active learning, in: 24th International Symposium on Research in Attacks, Intrusions and Defenses, 2021, 442–455.

[9] F. Barr-Smith, et al., Survivalism: Systematic analysis of windows malware living-off-the-land, in: 2021 IEEE Symposium on Security and Privacy (SP), 2021, 1557–1574.

[10] K. Ding, et al., LOLWTC: A deep learning approach for detecting living off the land attacks, in: 2023 IEEE 9th International Conference on Cloud Computing and Intelligent Systems (CCIS), 2023, 176–181.

[11] R. Harish, M. P. Swapna, Endpoint detection and response for fileless malware and LOLBin threats, in: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2024, 1–6.

[12] I. Ostroumov, et al., Modelling and simulation of DME navigation global service volume, Adv. Space Res. 68(8) (2021) 3495–3507.

[13] Z. Hu, et al., Statistical Techniques for detecting cyberattacks on computer networks based on an analysis of abnormal traffic behavior, Int. J. Comput. Netw. Inf. Secur. 12(6) (2020) 1–13.

[14] V. Kharchenko, I. Chyrka, Detection of airplanes on the ground using YOLO neural network, in: 2018 IEEE 17th International Conference on Mathematical Methods in Electromagnetic Theory (MMET), 2018, 294–297. doi:10.1109/MMET.2018.8460392

[15] I. Ostroumov, et al., A probability estimation of aircraft departures and arrivals delays, in: Computational Science and Its Applications, ICCSA, Lecture Notes in Computer Science, vol. 12950, 2021, 363–377. doi:10.1007/978-3-030-86960-1_26

[16] Z. Avkurova, et al., Models for early web-attacks detection and intruders identification based on fuzzy logic, Proced. Comput. Sci. 198 (2021) 694–699.

[17] O. Solomentsev, et al., Sequential procedure of changepoint analysis during operational data processing, in: 2020 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW), 2020, 168–171. doi:10.1109/MTTW51045.2020.9245068

[18] M. Iavich, et al., 5G Security function and its testing environment, in: Information Technology for Education, Science, and Technics, ITEST 2022, Lecture Notes on Data Engineering and Communications Technologies, vol. 178, 2022, 656–678.

[19] O. Solomentsev, et al., Data processing in case of radio equipment reliability parameters monitoring, in: 2018 Advances in Wireless and Optical Communications (RTUWO), 2018, 219–222. doi:10.1109/RTUWO.2018.8587882

[20] M. Iavich, et al., The novel system of attacks detection in 5G, in: Advanced Information Networking and Applications. AINA 2021. Lecture Notes in Networks and Systems, vol. 226, 2021, 580–591.