Malicious or Benign: Fine-tuning detection behaviors for Living-off-the-Land Windows Binaries

Peter Daniel[1 [0000−1111−2222−3333]], Emre Süren[1,2 [0000−0003−2356−8590]], and Teodor Sommestad[1,3 [2222−−3333−4444−5555]]

[1] KTH Royal Institute of Technology, Stockholm, Sweden
{emsuren,pedaniel,theodor}@kth.se
[2] Cybercampus Sweden, Stockholm, Sweden
emsuren@kth.se
[3] FOI, Linköping, Sweden
teodor.sommestad@foi.se

Abstract. The escalating trend of malicious actors employing Living-off-the-Land (LotL) binaries—legitimate system utilities already present in a system—as a defence evasion mechanism presents a significant challenge in cybersecurity. The dual usage of these techniques by both system administrators (sysadmins) and attackers has blurred the line between benign and malicious activities, leading to numerous false alarms in Intrusion Detection Systems (IDS), which diminishes their efficacy and overwhelms security teams.

This study investigated the challenges of accurately distinguishing between legitimate and adversarial use of LotL techniques, with a focus on how overlapping techniques impact IDS performance. To identify these intersecting LotL techniques, the research leveraged the MITRE ATT&CK® framework, the LOLBAS Project, and Sigma detection signatures, complemented by interviews and surveys with cybersecurity practitioners. A subset of administrative tasks resembling adversarial behaviours were automated and simulated in the Cyber Range And Training Environment (CRATE), while the Atomic Red Team framework was used to emulate malicious actions. Detection accuracy was evaluated using signature-based IDS tools: Snort, Sysmon, and Wazuh.

The findings demonstrate that despite using the same binaries and triggering identical signatures, subtle behavioural differences exist between benign and malicious usage of these binaries. Although all simulated activities were captured by the security monitoring tools, contextual analysis was consistently required to further distinguish between the two activities. Key recommendations include detection signature refinement and incorporating contextual analysis and anomaly detection to enhance IDS capabilities.

Keywords: Adversary emulation · Living-off-the-Land (LotL) · System administrator simulation · Cyber range · Behavior detection

1 Introduction

Accurately distinguishing cyberattacks from regular computer activities remains a critical challenge, largely because attackers often mimic the actions of authorized users and sysadmins. A prominent tactic among threat actors is the adoption of LotL tactics [6], where adversaries exploit native system binaries and utilities to blend seamlessly with normal system activity and network traffic, thereby evading detection [18]. For example, tools like PsExec [21] can be used legitimately by sysadmins for remote troubleshooting and system configuration, but also maliciously by attackers for lateral movement [1, 12]. This dual usage makes it arduous for defenders to discern malicious intent, and false alarms can be as critical as real attacks.

Traditional security solutions, including IDS and Security Information and Event Management (SIEM) tools, face significant challenges with LotL binaries [3] due to the overwhelming volume of false positives (FPs) they generate [20]. This study adopts the metrics of Benign Trigger (BT) for legitimate activities that meet alert conditions and True Alarm (TA) for actual threats, to provide a more nuanced understanding of detection performance. This research addresses the following questions:

- What are some of the most common LotL binaries used by both system administrators and threat actors?
- What are the overlaps and differences in the administrative and adversarial usage of LotL binaries? What types of false alarms are triggered due to these overlapping techniques?
- How effective are signatures in accurately differentiating between legitimate system administrator activities and malicious attacks, given the use of the same LotL binaries?

The purpose of this study is twofold: to demonstrate discernible distinctions in behaviour between legitimate users and adversaries even when using the same LotL binaries, and to identify weaknesses in IDS with default configurations to propose improvements for reducing FPs and enhancing detection accuracy. The project was conducted in collaboration with the Swedish Defence Research Agency FOI, with findings integrated into their Cyber Defence Exercises (CDXs) [17].

2 Background

2.1 Living-off-the-Land Binaries (LOLBins)

LOLBins are legitimate system utilities, binaries, processes, or tools native to operating systems (e.g., Windows, Linux, macOS) that are exploited by attackers on compromised systems to avoid detection [6]. This tactic allows adversaries to blend with legitimate activities and avoid deploying custom tools, making detection difficult. Examples of early LotL usage include APT29 [2] leveraging PowerShell and Windows Management Instrumentation (WMI). Various projects, such

as LOLBAS [13], GTFOBins [4], LOOBins [8], and LOLDrivers [7], catalogue these exploitable LotL techniques.

2.2 Threat Detection Methodologies

MITRE ATT&CK®: This framework [9], provides an extensive taxonomy for adversary behaviour, categorising Tactics, Techniques, and Procedures (TTPs) observed in real-world attacks. It was used to investigate common LotL binaries leveraged by APT groups and to map sysadmin and adversary LotL techniques.

LOLBAS Project: This project [13] documents LotL binaries, scripts, and libraries that can be exploited for malicious purposes, aiming to raise awareness and improve detection strategies. It includes Microsoft-signed files with unintended functionalities (e.g., hidden data in Alternate Data Streams, code execution) and offers ATT&CK® mappings and detection rules.

Sigma Rules: Sigma [15] is a generic detection format for SIEMs and a signature database with over 3000 threat detection rules. Its adaptable format allows for exchange of adversarial techniques across platforms and was employed in this study to identify LotL techniques that could trigger false alarms.

2.3 Related Work

Previous studies, such as Barr-Smith et al. [18], have investigated the prevalence of LotL techniques in malware, particularly APT malware, and their evasive capabilities against Antivirus (AV) solutions. This study builds upon that work by focusing on the overlap between administrative and adversarial use of LOLBins and assessing the efficacy of IDS and SIEM systems (rather than AV engines) in differentiating between legitimate administrative activities and intrusions. Qualitative studies by Alahmadi et al. [16] on false alarm fatigue in SOC analysts also informed this research, particularly in simulating real-world challenges for personnel in CDXs.

The most common LOLBins observed in the wild and frequently shared by both admins and attackers include [3]:

- **Execution:** PowerShell, Windows command shell, rundll32, regsvr32, mshta, MSBuild, te.exe, Wscript.
- **Lateral Movement and Persistence:** PsExec, WMI, WMIC.
- **Discovery:** ipconfig, tasklist, net, systeminfo, ping, nslookup.
- **Defense Evasion:** certutil, msiexec, netsh.

3 Methodology

This research adopted a mixed-methods approach, combining Literature Review (LR), qualitative research (semi-structured interviews and surveys), and empirical testing.

3.1 Qualitative Method

Semi-structured interviews and online surveys were conducted with industry experts, including Windows sysadmins and log analysts from four different organisations. The aim was to identify specific legitimate sysadmin activities involving LotL binaries [3] that could overlap with adversarial use. Ten LOLBins were selected for investigation based on literature review and their potential to induce false alarms: PsExec, msiexec, ssh, wmic, winget, wbadmin, certutil, ntdsutil, cmd, and cmdkey. Key qualitative findings indicated that:

- **Confirmed administrative usage:** SSH for proxying and remote access, PsExec with privilege elevation, Msiexec for offline software installation, and curl for downloads in automation scripts.
- **Mainly adversarial usage:** SSH for RDP tunnelling, CertUtil for downloading files, WMIC with XSL script formatting, and Cmdkey for credential listing.
- **Uncertain sysadmin use case:** PAExec and SSH ProxyCommand due to limited participant familiarity. These findings informed the selection of specific sysadmin behaviours for simulation.

3.2 Experimental Design and Data Collection

Empirical testing was conducted in CRATE [17] a simulated environment with Windows Virtual Machines (VMs) and Ubuntu VMs for remote services and security tools.

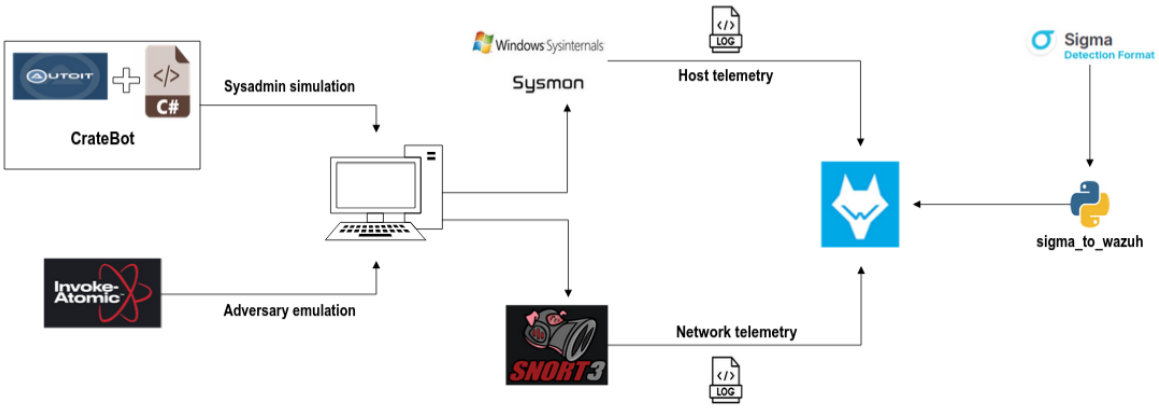

Fig. 1. System design

- Sysadmin behaviours were simulated using custom C# and AutoIt [5] automation scripts. Examples include quiet Wireshark installation with Msiexec, SSH proxying, and certificate management with CertUtil.
- Adversarial actions were emulated using the Atomic Red Team (ART) framework [14], including existing and newly developed atomic tests (e.g., SSH tunneling and proxying, greedy file deletion).
- Security Monitoring Tools: Snort [10], Sysmon [22], and Wazuh [11] were deployed to collect event logs and detect activities. Sysmon was configured with Olaf Hartong's modular template [19] to monitor system-level events, and Wazuh served as a central log aggregation engine for both Sysmon events and Snort logs.

4 Evaluation

4.1 Detection Efficacy: Overlap in Signatures

As hypothesised, the simulated benign and malicious activities, which both involved overlapping LotL techniques, triggered identical sigma signatures in Wazuh. A total of 12 sigma signatures were activated for each tested LotL technique by both legitimate administrative tasks and adversarial activities. All alerts generated by sysadmin activities were classified as Benign Triggers (BTs), while those from atomic tests were labelled True Alarms (TAs).

Sysmon and Wazuh Alerts: Sysmon consistently and reliably captured host-based events generated by both sysadmin and adversarial activities, providing extensive data such as command-line arguments, user, timestamp, and privilege escalation details (e.g., PsExec Local System Escalation). Wazuh effectively ingested these logs, correlated data, and visualised alerts.

Snort Detection: As a Network-based IDS (NIDS), Snort detected SSH and SMB communications and PsExec usage. However, its capability was limited; it could detect SSH traffic to standard or non-standard ports but could not inspect the invocation of specific commands (like ProxyCommand) due to SSH traffic being encrypted after the initial handshake. This limited Snort's ability to identify subtle behavioural distinctions within encrypted payloads.

4.2 Log Analysis: The Need for Contextualisation

Despite triggering identical detection signatures, contextual analysis proved crucial in differentiating benign from malicious activities.

- For example, Certutil was used for encoding in both scenarios: legitimately for certificate management by sysadmins, but maliciously to encode an executable for obfuscation.
- Msiexec was used by sysadmins for quiet installation of legitimate MSI packages (e.g., Wireshark), whereas adversaries leveraged it for executing MSI files containing embedded DLLs.

– SSH as a proxy was legitimately used by sysadmins to access isolated networks via an SSH gateway, while adversaries exploited it to execute binaries as a defence evasion tactic.

In production environments, additional Indicators of Compromise (IOCs) such as unusual timing (e.g., activity outside normal working hours), user roles (e.g., activity by unauthorized user accounts), and geographic anomalies can further aid in accurate distinction.

4.3 Reliability and Validity

The simulations were designed to reflect real-world scenarios, with sysadmin actions based on confirmed typical administrative behaviours identified through interviews/surveys. Adversarial emulations were aligned with the MITRE ATT&CK® framework, ensuring they replicate genuine threat actor TTPs observed in the wild. Consistent triggering of detection signatures across multiple test runs confirmed the reliability of the experimental setup.

5 Discussion

5.1 Findings Summary

Common LOLBins: PowerShell, Windows command shell, rundll32, MSBuild, PsExec, SSH, WMIC, certutil, msiexec, and netsh are prevalent binaries, with PsExec, Msiexec, SSH, CertUtil, PowerShell, and CMD commonly shared by both admins and attackers.

Overlaps and Differences: Overlapping behaviours include PsExec privilege escalation, SSH tunnelling, Msiexec quiet installation, CertUtil encoding, and CMD greedy deletion. Differences exist, such as sysadmins typically avoiding immediate execution of downloaded files, which is common adversarial behaviour. RDP tunnelling over SSH and file downloads via CertUtil were largely seen as adversarial.

Signature Effectiveness: Signature-based IDSs are prone to Benign Triggers (BTs) due to their reliance on static patterns. While Sysmon, Wazuh, and to a limited extent, Snort, alerted on suspicious activity, their output consistently required contextual analysis to determine intent, highlighting the limitations of signature-based detection in dynamic environments.

5.2 Implications and Recommendations

The findings imply that embedding behavioural and contextual analysis into detection mechanisms is crucial to significantly reduce BTs. Signature-based detection, while essential, has limited efficacy when used in isolation.

Key Recommendations for Improvement:

Detection Signature Refinement: Enhance existing Sigma rules by including additional filtering conditions, such as flagging potentially dangerous file types (e.g., executables, scripts, archives) in Certutil decoding operations.

Contextual Data Integration: Append rich event data fields from Sysmon (e.g., win.eventdata.user, win.system.computer, win.eventdata.UtcTime) to Wazuh custom rules to provide network defenders with better contextual information.

Incorporating Behavioural Analytics and Anomaly Detection: Strengthen detection by spotting deviations from baseline behaviours and detecting outliers, especially for novel attacks that signature-based methods struggle with.

Analyst Training: Emphasise training for security analysts to effectively perform contextual evaluations and make informed decisions about alerts.

Broader Scope for Future Research:

- Involve log analysts during the testing phase to observe their accuracy in distinguishing activities.
- Conduct interviews and surveys with larger and more diverse participant populations.
- Compare signature-based versus anomaly-based IDS.
- Include non-Windows binaries (Linux, macOS, cloud) and cloud environments for broader testing.
- Explore leveraging Large Language Models (LLMs), Machine Learning (ML), and Natural Language Processing (NLP) for contextual understanding, anomaly detection, and adaptive learning to improve IDS capabilities.

5.3 Research Contributions

This study has developed four new atomic tests to emulate specific adversarial techniques, including indirect command execution via SSH (T1202-6), SSH Proxying (T1021.004-3), SSH Remote Port Forwarding (T1021.004-4), and greedy file deletion via CMD (T1070.004-12). Additionally, a new Sigma rule titled "Suspicious Certutil Decoding" (Listing 6.1) has been developed and contributed to the SigmaHQ repository.

6 Conclusison

This project demonstrated that while host- and network-based detection systems can flag LotL behaviours used by both administrators and attackers, accurately distinguishing between the two fundamentally relies on network defenders conducting a thorough contextual evaluation. Current signature-based IDS, despite their effectiveness in detecting LotL activities, struggle to reliably differentiate benign from malicious actions, leading to an overwhelming volume of Benign Triggers. The work underscores the critical need for integrating contextual analysis, refining detection rules, and employing advanced behavioural detection methods to enhance the accuracy and efficiency of security systems and mitigate false alerts.

References

1. 8 LOLBINS Every Threat Hunter Should Know, https://bit.ly/3SFlDq8
2. APT29, IRON RITUAL, IRON HEMLOCK, NobleBaron, Dark Halo, StellarParticle, NOBELIUM, UNC2452, YTTRIUM, The Dukes, Cozy Bear, CozyDuke, SolarStorm, Blue Kitsune, UNC3524, Midnight Blizzard, Group G0016 | MITRE ATT&CK®, https://attack.mitre.org/groups/G0016/
3. ATT&CK® Navigator, https://mitre-attack.github.io/attack-navigator/layerURL=https://lolbas-project.github.io/mitre$_attack_navigator_layer.json$
4. GTFOBins, https://gtfobins.github.io/
5. Home, https://www.autoitscript.com/site/
6. Joint-Guidance-Identifying-and-Mitigating-LOTL, https://www.cisa.gov/sites/default/files/2025-03/Joint-Guidance-Identifying-and-Mitigating-LOTL508.pdf
7. LOLDrivers, https://www.loldrivers.io/
8. LOOBins, https://www.loobins.io/
9. MITRE ATT&CK®, https://attack.mitre.org/
10. Snort - Network Intrusion Detection & Prevention System, https://www.snort.org/
11. Wazuh, https://github.com/wazuh/wazuh/
12. Quantum Ransomware (Apr 2022), https://thedfirreport.com/2022/04/25/quantum-ransomware/
13. LOLBAS-Project/LOLBAS (Oct 2024), https://github.com/LOLBAS-Project/LOLBAS, original-date: 2018-06-08T22:11:05Z
14. redcanaryco/atomic-red-team (Oct 2024), https://github.com/redcanaryco/atomic-red-team, original-date: 2017-10-11T17:23:32Z
15. SigmaHQ/sigma (Oct 2024), https://github.com/SigmaHQ/sigma, original-date: 2016-12-24T09:48:49Z
16. Alahmadi, B.A., Axon, L., Martinovic, I.: 99% false positives: a qualitative study of {SOC} analysts' perspectives on security alarms. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 2783–2800 (2022)
17. Almroth, J., Gustafsson, T.: CRATE Exercise Control – A cyber defense exercise management and support tool. In: 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). pp. 37–45 (Sep 2020). https://doi.org/10.1109/EuroSPW51379.2020.00014, https://ieeexplore.ieee.org/document/9229649
18. Barr-Smith, F., Ugarte-Pedrero, X., Graziano, M., Spolaor, R., Martinovic, I.: Survivalism: Systematic Analysis of Windows Malware Living-Off-The-Land. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 1557–1574 (May 2021). https://doi.org/10.1109/SP40001.2021.00047, https://ieeexplore.ieee.org/abstract/document/9519480, iSSN: 2375-1207
19. Hartong, O.: olafhartong/sysmon-modular (Feb 2025), https://github.com/olafhartong/sysmon-modular, original-date: 2018-01-13T21:20:59Z
20. Jaber, A.N., Rehman, S.U.: Fcm–svm based intrusion detection system for cloud computing environment. Cluster Computing **23**(4), 3221–3231 (2020)
21. markruss: PsExec - Sysinternals (Mar 2023), https://learn.microsoft.com/en-us/sysinternals/downloads/psexec
22. markruss: Sysmon - Sysinternals (Jul 2024), https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon