Silent Intruders: Understanding Living-off-the-Land Techniques, Threats, Countermeasures and Emerging Solutions

Nicholas Kavadias
Business Justice & Behavioural Sciences
Charles Sturt University
Sydney, NSW, Australia
February, 2024
nick+research@kavadias.org

***Abstract*—This research addresses the sophisticated cyber threat technique Living Off the Land (LOTL), where adversaries utilise legitimate system tools for malicious purposes. With LOTL techniques blending seamlessly into normal system operations, they present a formidable challenge to cyber security. This study explores the relationship of LOTL with fileless malware and looks at what LOTL techniques are and the threat actors that use them. It examines the anatomy of cyber attacks to understand where LOTL techniques are utilised within attack phases. It conducts a comprehensive survey of the latest countermeasures methods that can be used to defend against LOTL attacks. The research discusses the difficulty in defending against LOTL attacks by organisations. The research aims to bolster cyber security defenders' knowledge base and awareness significantly, enhancing organisational resilience against sophisticated threats that are exploiting LOTL techniques. The findings seek to contribute to developing more effective detection methodologies, thereby fortifying defences against the stealthy and evolving nature of LOTL attacks.**

***Index Terms*—Living of the Land (LOTL), Fileless malware, Advanced Persistent Threat (APT), Hacking, Cybersecurity,**

I. Introduction

In the rapidly evolving cyber security threat landscape, adversaries continually seek innovative ways to bypass traditional defences. Living off the Land (LOTL) attacks represent a sophisticated class of cyber threats wherein attackers utilise legitimate, system tools to conduct malicious activities, thus camouflaging their actions within normal system processes and user activities. This approach makes LOTL attacks particularly stealthy and challenging to detect, as they blend in with legitimate operations, bypassing traditional system security measures.

Recent statistics underscore the growing prevalence and sophistication of LOTL attacks. According to CrowdStrike [1], 71% of all detections were malware-free in 2022, up 9% on the previous year. The Australian Signals Directorate (ASD) [2] reveals in their latest threat report a significant surge in Advanced Persistent Threats (APTs) against Australian critical infrastructure, citing 143 incidents, an increase of 51% over the previous year, further highlighting the shift towards more sophisticated, stealthy attack methods that evade traditional detection mechanisms.

This study looks at LOTL techniques, examining the nature of LOTL attacks, the concept of LOLBins, and how these attacks are executed. It contrasts LOTL techniques with fileless malware, outlining a hierarchy of attack methods. The paper discusses the variety of threat actors employing LOTL strategies and their challenges to cyber security defences.

This study is particularly interested in how LOTL cyber threats are defined, the anatomy of cyberattacks utilising LOTL techniques, and the latest countermeasures devised to thwart such attacks. The research offers insights into the detection and defence mechanisms that can enhance organisational resilience against such tactics.

Areas outside the scope of this research include the use of LOTL techniques against non-Windows platforms despite acknowledging their relevance in the broader discussion of cyber security threats. Furthermore, this study does not focus on a particular threat actor associated with LOTL, such as APTs or threat types, such as ransomware. However, it offers insights into the where, why and how these threat actors use LOTL techniques.

The overall findings from this review indicate a growing prevalence of LOTL attacks by all threat actors and the inadequacy of traditional detection methods in stopping these threats. Although LOTL techniques pose a formidable challenge, modern detection methods, like behavioural anomaly detection (BAD) can improve organisational defences against these stealthy attacks.

II. Methodology

The methodology process undertaken for this research is outlined in Figure 1.

The project questions identified were categorised into two groups, as detailed in Table 1. **Group A** questions focused on the understanding and background of LOTL threats, and **Group B** questions on the countermeasures and challenges in defending against LOTL threats.

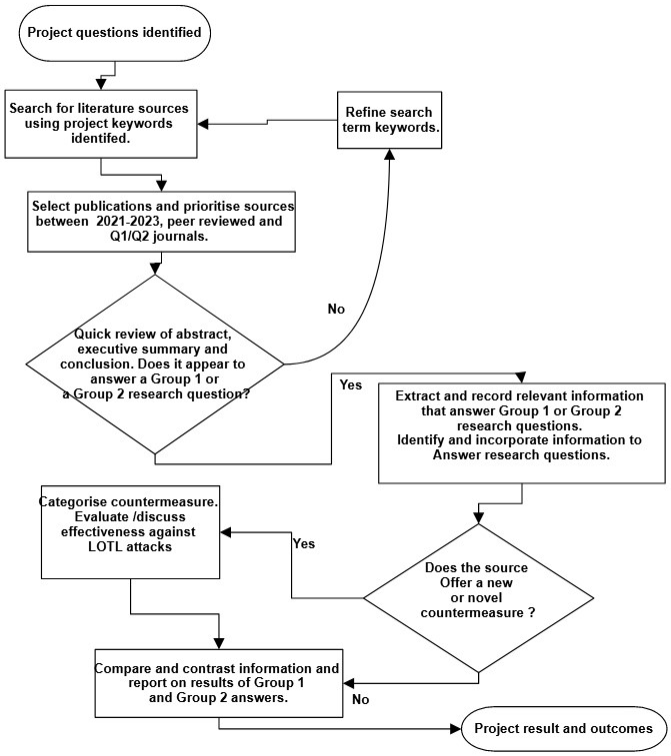

Figure 1. Flowchart summarising literature review methodology used in finding sources and synthesising results.

Understanding and Background (Group A)	**Countermeasures and Challenges in Defence (Group B)**	**Search Terms**
QA1 – What is a LOTL attack? *QA2*– What is a fileless malware? *QA3* - What kind of cyber threats use LOTL? *QA4* - What phases of a cyberattack can use LOTL techniques?	*QB1*- Why are threat actors increasingly using LOTL techniques? *QB2*- What are the limitations and challenges in defending against LOTL attacks? *QB3* - What countermeasures can be used to defend against LOTL? *QB4* - Are there novel or emerging technologies that can help defend against LOTL attacks?	"living off the land" LOTL LOLBin LOLBas "LOL Binaries" "fileless malware" "fileless APT" "malware evasion" "malware stealth" "APT evasion" "APT stealth" "APT detection" "Advanced Persistent Threat"

Table I
RESEARCH QUESTIONS AND SEARCH TERMS

.

The literature search used multiple academic platforms such as Google Scholar, Primo Search, IEEE Xplore, Springer and Wiley. The search terms were expanded to terms that may discuss LOTL, including broader terms such as "fileless malware" or specific threats such as "APT evasion". The search terms used are listed in Table 1. Notably, the search did not explicitly include the term "GTFOBins", often used to describe the LOTL binaries for Linux and Unix platforms. Given the cutting-edge nature of the research, there were limited relevant results from peer-reviewed journals, even with the expanded search terms. The scope of the research was expanded to incorporate conference proceedings from industry conferences, large software vendor reports and government intelligence organisations that have addressed LOTL cyber threats. Additionally, specialist vendor and community websites offering technical details on LOTL threats were consulted. A total of thirty-two sources were condensed down to fifteen, which answered Group A and/or Group B questions. Information was identified, extracted, and categorised. Sources were classified as answering Group A or B research questions and whether they offered or researched a novel approach in a defence or mitigation strategy in defending against LOTL. Results were critically evaluated, and implications of the findings were assessed and discussed.

III. RESULTS

Of the fifteen sources selected for research, eight were journal articles, four were conference proceedings, one was a governmental report, one was a vendor report, and one was a cybersecurity community project. Seven sources answered research questions on the definition and understanding of LOTL techniques. Fourteen of the sources discussed the challenges in defending against LOTL or the countermeasures that could be used to defend against LOTL techniques. Five of the sources offered novel methods in defending against LOTL attacks, either as proposed theories or as quantitative research. The sources are summarised by research questions answered in Table II.

A. What is a LOTL attack?

Traditionally, the phrase "living off the land" describes the practice of subsisting on natural resources through farming and hunting. In cyber security, this concept is analogously applied to threat actors exploiting installed binaries, scripts and tools on target systems to carry out malicious objectives. The phrase and the acronym LOTL were first popularised at the hacker conference DerbyCon in 2013, along with related terms like LOLBins and LOLBAS – acronyms for *L*iving-*o*ff-the-*l*and *Bin*aries *A*nd *S*cripts [3]. LOLBAS are the executable binary tools on computer systems and scripts run by code interpreters that are used to achieve malicious purposes.

Following DerbyCon, the LOLBAS Project commenced, a Github [4] project aiming to compile a comprehensive catalogue of exploitable Windows binaries, scripts, and libraries[3]. The project serves as an educational tool, aiding in developing detection and defence strategies. It documents the

Sources	Group A	Group B	Novel defence?
Barr-Smith et al.	✓	✓	✗
Bhardwaj et al.	✗	✓	✓
Boros et al.	✗	✓	✓
CISA Report	✓	✓	✗
Fan, Liu and Perigo	✗	✓	✗
Lee et al.	✓	✗	✗
Liu et al.	✓	✓	✗
LOLBAS Project	✓	✓	✗
Microsoft Threat Intell. Report	✓	✓	✗
Najafi et al.	✗	✓	✗
Ning al.	✗	✓	✗
Ongun et al.	✗	✓	✓
Salim et al	✗	✓	✓
Sharma et al.	✓	✓	✗
Tsai et al.	✗	✓	✓
TOTAL: 15	**7**	**14**	**5**

Table II
SOURCES CATEGORISED BY RESEARCH ANSWERS PROVIDED

.

LOLBin and specifies the stage that the LOLBin could be used by mapping it to the attack stages in the MITRE ATT&CK framework [5]. This framework is extensively used for cataloguing the techniques of threat actors, gaining widespread adoption in the cyber security industry. A parallel initiative also exists to document LOTL Unix binaries [6].

The LOLBAS Project delineates a strict criterion for listing a binary as taking advantage of a LOTL technique: it must be a Microsoft-signed file native to the operating system or downloaded from Microsoft. It must have unexpected (or undocumented) functionality, and the functionality must be useful to a threat actor. Useful functionality for a threat actor implies that it corresponds to a specific stage and tactic within the MITRE ATT&CK Framework.

In the analysis by Barr-Smith, Ugarte-Pedrero, Graziano *et al.* [7] of 31 million Windows malware samples for LOTL techniques there was no delineation of binaries for expected functionality versus unexpected (malicious) purposes.

Sharma, Gupta, Singh *et al.* [8], in discussing the evolution of ATPs, conflate the term ”fileless malware” with LOTL. The paper describes fileless malware as coming under the category of LOTL. The study calls LOLBins techniques described by sources [3],[9] and [10] by a phrase unique to the paper, calling them “Windows Platform Techniques”.

B. What about fileless malware?

Liu, Peng, Zeng *et al.* [10] provides a structured definition of fileless malware, classifying LOTL techniques as a specific sub-type of fileless malware. They define fileless attacks as being of three types: *Memory-based* attacks that rely on vulnerability exploitation, memory-resident malware and process injection. *Service-based* attacks rely on Windows services such as the Windows Registry, Scheduled Tasks or Alternative Data Streams (ADS), and finally, *LOTL-based* attacks that include malicious document attacks, script attacks, and LOLBins-based attacks. In their definition, not all ”fileless attacks” are actual file-less, refer to Figure 2.

Attack Type	Techinique	Require file carrier?
Memory-based	Vulnerability exploitation	○
	Memory resident malware	○
	Process injection	○
Service-based	Registry resident attack	◑
	Scheduled task	◑
	ADS attack	●
LotL-based	Malicious document attack	●
	Script attack	◑
	LoLBins-based attack	◑

Classification of fileless attack. “●”, “◑”, “○” means “Yes”, “Maybe”‘, “No”.

Figure 2. Categorisation of fileless attacks, source: Liu, Peng, Zeng *et al.* [10]

Lee, Shim, Cho *et al.* [11], in their study, do not differentiate between fileless malware, LOTL or LOLBins, classifying them all simply as ”fileless attacks”. However, they excluded from their definition any attack that requires files on the file system, even temporarily.

C. What kinds of cyber threats use LOTL techniques?

Analysis by [7] of large datasets of all kinds of malware found that 9.41% made use of LOLBins, as well as 26.26% of APT malware, with more advanced APTs such as Hurricane Panda and Lazarus making use of LOLBins 100% of the time. The study also found popular ransomware families Cerber and Gandcrab used LOTL techniques, which are especially useful for silently deleting backups to prevent system recovery.

Reports [9],[12] that analysed APT threat actor Volt Typhoon highlighted how this Chinese state-sponsored threat used LOTL techniques almost exclusively to infiltrate critical infrastructure. The threat actor using *wmic.exe*, *ntdsutil.exe*, *netsh.exe*, and *powershell.exe* to perform their objectives and blend in with normal system and network activities.

Other papers [10],[8],[13] that focused on LOTL used by APTs discussed ways threat actors use LOLBins in ”hands-on keyboard” attacks, after initially gaining entry by credential theft through phishing or systems with unpatched vulnerabilities.

D. What phases of a cyber attack can use LOTL techniques?

A comprehensive analysis by Sharma, Gupta, Singh *et al.* [8] included a detailed comparison of APT stages across different models, illustrating the versatility and prevalence of LOTL tactics throughout the cyber attack lifecycle. Figure 3 from Sharma’s study visually represents this comparison,

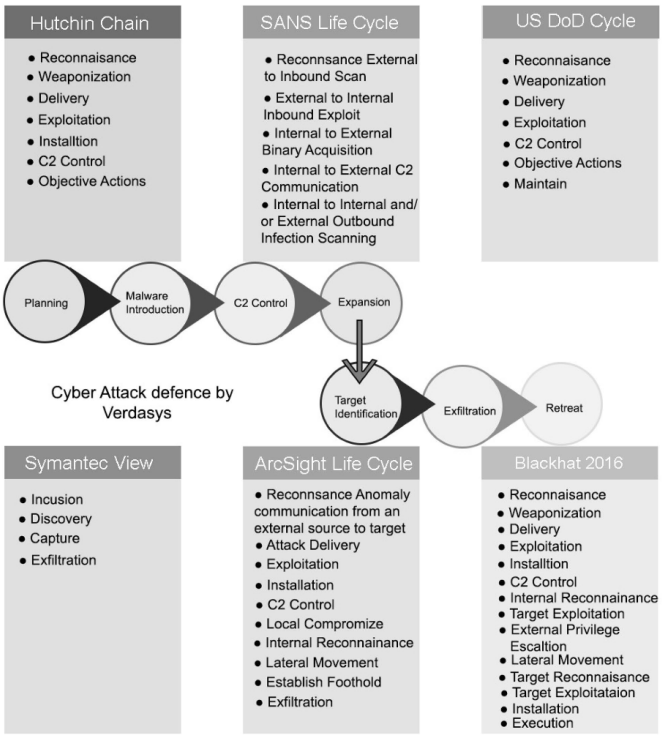

Figure 3. Comparison of APT attack frameworks, source: Sharma, Gupta, Singh *et al.* [8]

	EVASION		ATTACK		COLLECTION		
Malware	Persistence	Defense Evasion	Privilege Escalation	Impact	Credential Access	Discovery	Collection
Poweliks	4	4	2	0	0	1	0
Rozena	3	4	2	0	0	1	0
Duqu 2.0	1	1	1	0	0	0	0
Kovter	4	5	2	0	1	4	0
Petya	1	2	1	0	0	0	0
Sorebrect	1	2	1	0	0	2	0
WannaCry	6	4	2	0	0	2	0
Magniber	2	2	2	0	0	1	0
Emotet	3	4	2	0	0	0	0
GandCrab	1	2	1	0	0	1	0
Totals	26	30	16	0	1	12	0

Figure 4. Coloured data bar summary of malware attack tactics mapped to the MITRE ATT&CK framework from research by Lee, Shim, Cho *et al.* [11]

highlighting the alignment of various APT stages with the phases outlined in the MITRE ATT&CK framework.

In Lee, Shim, Cho *et al.* [11], an empirical study mapped the phases of LOTL techniques in well-known malware to seven of the twelve MITRE ATT&CK Framework tactics showcasing specific examples of LOTL techniques in action. The most prevalent use of LOTL techniques was in Defence Evasion, Persistence and Privilege Escalation, as summarised by the coloured data bar table synthesised from data in Lee's research (Figure 4).

The *LOLBAS Project* [3] has a comprehensive listing of 198 Windows LOLBins mapped to the MITRE ATT&CK Framework on their project website and accepts community submissions to ensure the list is kept up-to-date. From the binaries listed there, 187 different LOTL tactics were in use across all twelve MITRE ATT&CK phases, the largest number of techniques catalogued being for the Defence Evasion tactic.

E. What are the challenges and countermeasures in defending against LOTL?

In the domain of cyber security, effectively managing risks involves implementing controls tailored to mitigate specific threats. This research reviewed sources that identified and discussed controls that could be put in place to prevent or detect LOTL threats.

Liu, Peng, Zeng *et al.* [10] addressed the challenges inherent in AV methods for detecting attacks. They discussed the ineffectiveness of signature-based and rule-based detection methods due to their inability to identify unknown attacks and heavy reliance on expert knowledge and extensive rules databases.

In 2023, significant attention was drawn to LOTL techniques after state-sponsored APTs targeted critical infrastructure. CISA's report [12] on this threat identified LOTL attacks. It stated that many of the commands used by ATPs and listed as indicators of compromise (IOCs) might, in fact, be normal system behaviour and that defenders should investigate further before assuming compromise.

1) Behavioural Anomaly Detection: Ning, Bu, Ju *et al.* [14] surveyed behavioural anomaly detection (BAD) research that offered novel methods for improving the accuracy of detecting LOTL commands, i.e., being able to predict whether a command issued to a system was benign or malicious by only looking at the binary and command line parameters of the execution. They classify these types of detection methods by the technology behind them: pattern matching, natural language processing (NLP), and machine learning (ML). See Figure 5 for their taxonomy of BAD methods.

Pattern matching is the most traditional method for detecting LOTL techniques. The open-source pattern matching tool *YARA - The pattern matching swiss knife for malware researchers* [15] is cited in papers [10],[11], [14], and [16] as the pattern matching detection tool for cyber security and is often used to capture the detection rules for specific threats. Yara rules are commonly published in threat intelligence reports as it was found in [12].

Research by Ongun, Stokes, Or *et al.* [17] showed that BAD using ML under a supervised learning training method was an effective detection method, achieving an accuracy (F1 score) of 96%. Supervised learning is a method where commands that could not be classified as safe or malicious were forwarded to a cyber security analyst to determine. The process aids in the precise categorisation of these LOTL techniques and enhances the model's learning capability by incorporating expert feedback. The ML training leveraged large datasets from real-life implementations of the EDR product Microsoft Defender for Endpoint[18].

Boroş, Cotaie, Stan *et al.* [19] used a novel combination of cybersecurity experts to help pattern match and classify datasets of risky commands before ML training for BAD.

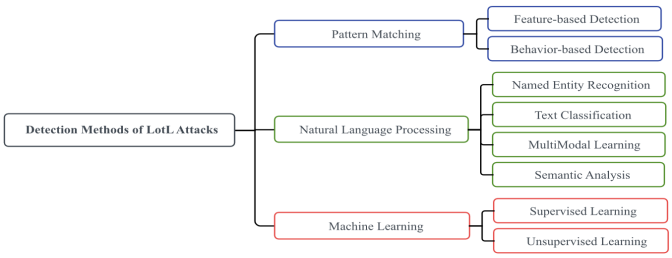

Figure 5. Detection method taxonomy by technology, Source: Ning, Bu, Ju *et al.* [14].

Unlike [17], Boros used a combination of manual labelling and ML Random Forest classification, achieving an accuracy score of 95%. Again, the research relied on large datasets of real-world EDR data.

Research by Tsai, Lin, He *et al.* [16] combined deep learning and Natural Language Processing (NLP), specifically for de-obfuscating and detecting malicious Powershell commands. Powershell, a script interpreter for Windows, is a major source of LOTL attacks. The research achieved an F1 score of 98.5% for malicious Powershell commands.

2) Other countermeasures: **Threat Hunting**: Bhardwaj, Kaushik, Alomari *et al.* [20] offered a novel Behavior-Based Threat Hunting (BTH) framework designed to counter APTs, fileless malware, and LOTL techniques. The process is a non-technical procedural process that can be followed by threat-hunting teams in organisations. The framework emphasises proactive, behavioural analysis and threat-hunting strategies over traditional, reactive security measures. The novel BTH method incorporates threat intelligence and situational awareness to detect cyber threats. It prioritises behavioural patterns, utilises threat intelligence data, and uses an analytics-driven threat-hunting process.

Provenance Graphing : The approach presented in Najafi, Pünter, Cheng *et al.* [21] combines machine learning with graph theory and data analytics techniques. It integrates the analysis of host-level system events/logs into what they call a Heterogeneous Information Network (HIN), applying graph-based inference algorithms to detect malicious activities based on information from EDR and Security Information and Event Management (SIEM) data. This method leverages ML for analysing patterns and relationships within the data but extends beyond traditional ML by incorporating graph mining to infer the maliciousness of entities in a complex network of interactions. Researchers achieved an F1 score of 83%, and the most promising was that their technique could detect new threats of previously unknown attacks without re-training.

Threat Triage : Unique research by Fan, Liu and Perigo [22] highlighted the resource problem of EDR detection that needs to monitor all system processes, showing that APTs can easily overwhelm a system's computing resources in an attack. To solve this, they propose an innovative approach that integrates a neural network system, enhancing detection capabilities by utilising multiple signals and creating provenance graphs, meaning that with limited system resources an EDR can triage potential threats for analysis based on their risk level and thus reducing overall impact on computing resources.

IV. Discussion

A. No consensus of definitions for LOTL, LOLBin or fileless malware

The results clearly show that there is a lack of a standard definition of what LOTL attacks, LOLBins, and fileless malware are within the cyber security and academic community. The results illustrate the breadth of interpretations across academic and cyber security communities, suggesting that the lack of consensus stems from the evolving landscape of cyber threats.

Whilst the LOLBAS Project [3], has a discreet and narrow definition of what a LOTL technique is for a LOLBin to be included in its project, i.e. that a system tool needs to have a secondary or unexpected usage to be listed, it is somewhat lacking in the practical sense that defenders often need to defend against threats which use system tools for their intended purpose but are unauthorised. A good example of this is the *vssadmin.exe* tool on Windows that has documented functionality for deleting file system backups [23]. This command is weaponised by ransomware malware to prevent system recovery[7]. This tool is not considered a LOLBin by the LOLBAS Project, but clearly, it is a tool that should be controlled by defenders to prevent abuse.

The research by Liu, Peng, Zeng *et al.* [10] attempted to provide a comprehensive analysis, differentiating categories of "fileless attacks". They defined fileless attacks into three categories (Figure 2), which, by their own definition, two of the categories use files. By contrast, Lee, Shim, Cho *et al.* [11] defined fileless attacks to mean completely fileless attacks whose malicious payloads execute in memory without relying on files stored on disk. Furthermore, the term "Windows Platform Techniques" used by [8] described what [10] referred to as Service-based attacks. i.e., attacks that exploit Windows operating system features like the Windows Registry or the Scheduled Tasks service for malicious means. These service-based attacks are often abused using built-in command-line tools such as *reg.exe, regsrv.exe, and regsrv32.exe* for the Windows Registry and *sctasks.exe and at.exe* for the Windows Task Scheduler service, explaining why other sources [9] and [10] classify these as LOTL attacks.

These discrepancies in definition can hinder collaborative efforts to develop effective defence mechanisms, leading to fragmented understandings and approaches in defending against attacks.

To address these challenges and in an attempt to harmonise these disparate definitions, it is more useful to see LOTL techniques as a hierarchy (Figure 6) having a large base encompassing a wide range of techniques, and having a much more narrow definition of attacks at the apex. At the bottom of the hierarchy is the wide Liu definition of fileless malware. The only true fileless attack is a resident memory attack, as defined by Lee. This is a practical boundary to draw for a distinction

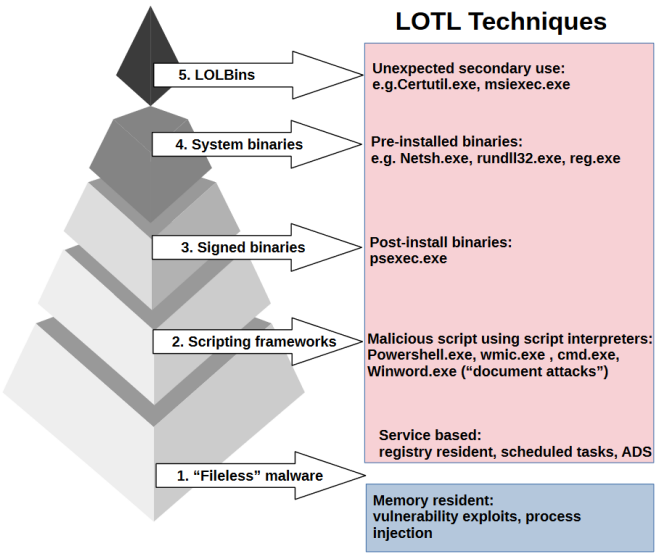

Figure 6. The hierarchy of LOTL techniques.

between what is and what is not a LOTL attack. Memory resident attacks are most commonly caused by unpatched software bugs that allow an attacker to execute arbitrary code, i.e. the Initial Access stage of MITRE ATT&CK. Further up in the hierarchy we have service-based attacks. Service attacks are indeed LOTL attacks as threat actors leverage Windows Services for malicious intent. Furthermore, we categorise service-based and exploited scripting frameworks together with document-based attacks, as document attacks work by leveraging scripting languages like JavaScript or Visual Basic Applications (VBA) macros embedded in the documents. Next in the hierarchy are signed binaries; these are binaries that are not installed on the operating system but are signed by the OS vendor e.g. Microsoft, meaning the binaries will often run even if application whitelisting is in place. All system binaries are signed binaries, which means there are fewer of them, and thus, they are placed higher in the hierarchy. Signed and system binaries exist for legitimate purposes and may not have unexpected or secondary functionality but can nonetheless be exploited by a threat actor in control of a system. Finally, at the peak of LOTL are signed binaries that have unexpected functionality, which is the requirement for a LOLBin qualify for listing as a LOLBAS on the project. An example of this is *certutil.exe*, a tool intended to be used for managing a computer's certificate store but has malicious secondary uses such as its ability to download arbitrary files from the internet, and encode/decode files to and from Base64 [24].

B. Inadequacy of existing controls

1) Antivirus tools are not designed to detect LOTL threats: The biggest challenge with LOTL techniques is detecting and stopping malicious activity, which is often indistinguishable from legitimate, authorised activity. Almost all of the journal sources reviewed in this research acknowledged the problem of traditional file-based signature methods used by AVs in failing to detect LOTL techniques. Traditional malware was compiled code that spread by self-copying, hence the term virus. The detection of malicious binaries in AV products is based on the hashing of files to create signatures; that is where AV products compare a file signature to a store of known bad signatures. More advanced AVs use heuristic detection to look at code execution, i.e. code that looks risky and could perform malicious actions. Threat actors, knowing this, have been drawn to using LOTL techniques because it is impossible for AV to detect malicious behaviour of tools that are part of the operating system and are signed OS vendors like Microsoft. This inadequacy of AV is highlighted by the research results in [7]. Researchers tested LOTL techniques of five different LOLBins (*ftp.exe, mshta.exe, rundll32.exe, regsrv32.exe, bitsadmin.exe*) against the top ten AV products. Two of the ten AV products detected two out of five attacks, with AV detection being 15% overall across all top ten AV products. Even after the researchers disclosed the LOTL attacks bypassing AV to vendors and re-tested the products nine months later, the overall AV detection rate using the same attacks only improved to 42%, and all AVs failed to detect at least one of the LOTL attacks.

2) Hiding in plain sight: The growing sophistication of cyber threats has seen a marked increase in the adoption of LOTL techniques because LOTL techniques allow threat actors to hide in plain sight. State-sponsored APTs targeting critical infrastructure[12],[9] have notably leveraged these tactics because malicious actions by system tools are indistinguishable from normal behaviour. Detection tools, either AV or EDR cannot derive the intent of an action, complicating detection efforts significantly.

LOTL techniques used by threat actors are predominantly used for Defensive Evasion. See Figure 7 for a summary showing the total techniques organised by the MITRE ATT&CK Stage from the LOLBAS Project[3]. Of the 187 techniques reported by LOLBAS, 34 of them are for Defensive Evasion, and most of these are in the category of proxied execution. This method is very useful in bypassing security controls on a system that only allows signed binaries to run. Proxied execution means that a threat actor can run code via a process that is allowed, i.e. the signed binary, a process which would otherwise not be allowed to run alone.

In [12] and [9] reports which covered the LOTL techniques used by APT Volt Typhoon, the reports made it clear that even if commands detailed in the reports were found to have been run on systems, if files that were Indicators of Compromise (IOCs) were discovered such as backup Active Directory backup files *NTDIS.dit*, they could be false positives of malicious activity. As the commands and files used by threat actors could also be run and exist for legitimate use by a systems administrator. Again, this points to the malicious activity's context and not the activity itself being the threat.

MITRE ATT&CK Stage	Total Techniques
1. Initial Access	9
2. Execution	10
3. Persistence	18
4. Privilege Escalation	13
5. Defensive Evasion	34
6. Credential Access	16
7. Discovery	26
8. Lateral Movement	9
9. Collection	15
10. Command and Control	16
11. Exfiltration	8
12. Impact	13
TOTAL	187

Figure 7. Summary of LOLBAS Project LOLBin techniques mapped to MITRE ATT&CK.

C. The advantages and disadvantages of newer detection methods

Pattern Matching tools such as Yara, although useful in detecting LOTL IOCs, are hampered by the need for constant updates of threat indicators and require significant resource allocation to scan resources for IOCs. There can also be high false positive rate, as the precision of the detection is only as good as the rule.

Machine Learning offers strong adaptability and scalability as illustrated in the results of [19] and [17]. Despite its potential, the requirement for extensive training data creates a problem, meaning there needs to be threat data to be able to train the models to detect new threats. [14] also discussed the possibility of poisoning training data. This could mean creating false positives to make detection less reliable and trustworthy or using new techniques sparingly and obfuscating them continually to avoid detection.

Natural Language Processing used in [16] emerges as a flexible tool for detecting new malware types, using advanced techniques like neural networks. Yet, the large model sizes and the complexity of feature extraction pose considerable challenges, potentially limiting its practical deployment in rapidly changing security environments.

V. Future Work

The evolving nature of LOTL attacks necessitates ongoing research and development of innovative countermeasures. Future work should focus on several key areas to enhance our defensive capabilities against LOTL techniques. Firstly, developing more sophisticated detection mechanisms that leverage AI and ML should be prioritised. These technologies, particularly those capable of BAD, hold promise for identifying subtle anomalies that are indicative of LOTL activities.

The novel LOTL detection research [17],[19] and [16] all focused on looking at individual commands in trying to derive whether the command was benign or malicious. Those papers' future research sections all pointed out the limitations of not knowing the context of previous commands and that malicious commands often happen in groups. Research in deriving intent from commands used based on their context would be valuable in improving BAD.

Additionally, exploring provenance graphing in detecting malicious activities showed a novel approach in [21] and [22]. The application of provenance graphs to detect malicious LOTL commands could be a promising area of research.

Lastly, collaboration between academia, industry, and government agencies is essential for developing a comprehensive knowledge base and the standardisation of language around naming the kinds of threats being encountered in LOTL. My research shows that more formal definitions for the class of threats LOTL creates need to be developed.

VI. Conclusion

This paper has explored the landscape of LOTL cyber attacks, their mechanisms, and the challenges in defining, detecting, and mitigating LOTL threats. By reviewing what defines a LOTL attack, the threat actors that use LOTL attacks, and the countermeasures for defending against them, it becomes evident that traditional cyber security defences are often inadequate against the stealth and sophistication of LOTL techniques. The findings underscore the need for advancing detection technologies, incorporating BAD beyond pattern matching rules to encompass ML, NLP, and provenance graphing to effectively discern and counteract threats.

The research highlights a pivotal shift in the cyber security paradigm, where understanding the context of system commands and integrating behavioural detection is necessary to identify and stop LOTL attacks. Moreover, the study calls for a unified approach towards defining LOTL attack terminology and advocating for further collaboration across academia, industry, and governments.

As we look towards the future, it is clear that the battle against LOTL attacks will require technological innovation and a nuanced understanding of threat and evolving attack strategies. Ultimately, our ability to adapt and improve our defences will be crucial in defending our organisations against these silent intruders.

References

[1] CrowdStrike. '2024 global threat report — crowdstrike.' (21 Oct. 2023), [Online]. Available: https://www.crowdstrike.com/global-threat-report/ (visited on 02/02/2024).

[2] Australian Signals Directorate (ASD). 'Asd cyber threat report 2022-2023.' (14 Nov. 2023), [Online]. Available: https://www.cyber.gov.au/about-us/reports-and-statistics/asd-cyber-threat-report-july-2022-june-2023 (visited on 09/12/2023).

[3] 'Lolbas project.' (), [Online]. Available: https://lolbas-project.github.io/ (visited on 05/01/2024).

[4] 'Github: Let's build from here.' (), [Online]. Available: https://github.com/ (visited on 31/01/2024).

[5] MITRE Corporation. 'Mitre att&ck®.' (), [Online]. Available: https://attack.mitre.org/ (visited on 21/01/2024).

[6] 'Gtfobins project.' (), [Online]. Available: https://gtfobins.github.io/ (visited on 06/01/2024).

[7] F. Barr-Smith, X. Ugarte-Pedrero, M. Graziano, R. Spolaor and I. Martinovic, 'Survivalism: Systematic analysis of windows malware living-off-the-land,' 1 May 2021. DOI: 10.1109/sp40001.2021.00047. [Online]. Available: https://doi.org/10.1109/sp40001.2021.00047.

[8] A. K. Sharma, B. B. Gupta, A. K. Singh and V. K. Saraswat, 'Advanced persistent threats (apt): Evolution, anatomy, attribution and countermeasures,' *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 7, pp. 9355–9381, 6 May 2023. DOI: 10.1007/s12652-023-04603-y. [Online]. Available: https://doi.org/10.1007/s12652-023-04603-y.

[9] Microsoft Threat Inteligence. 'Volt typhoon targets us critical infrastructure with living-off-the-land techniques.' (17 Oct. 2023), [Online]. Available: https://www.microsoft.com/en-us/security/blog/2023/05/24/volt-typhoon-targets-us-critical-infrastructure-with-living-off-the-land-techniques/.

[10] S. Liu, G. Peng, H. Zeng and J. Fu, 'A survey on the evolution of fileless attacks and detection techniques,' *Computers & Security*, vol. 137, p. 103 653, 1 Feb. 2024. DOI: 10.1016/j.cose.2023.103653. [Online]. Available: https://doi.org/10.1016/j.cose.2023.103653.

[11] G. Lee, S. Shim, B. Cho, T. Kim and K. Kim, 'Fileless cyberattacks: Analysis and classification,' *Etri Journal*, vol. 43, no. 2, pp. 332–343, 17 Dec. 2020. DOI: 10.4218/etrij.2020-0086. [Online]. Available: https://doi.org/10.4218/etrij.2020-0086.

[12] CISA. 'People's republic of china state-sponsored cyber actor living off the land to evade detection — cisa.' (24 May 2023), [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa23-144a.

[13] D. T. Salim, M. Singh and P. Keikhosrokiani, 'A systematic literature review for apt detection and effective cyber situational awareness (ecsa) conceptual model,' *Heliyon*, vol. 9, no. 7, e17156, 1 Jul. 2023. DOI: 10.1016/j.heliyon.2023.e17156. [Online]. Available: https://doi.org/10.1016/j.heliyon.2023.e17156.

[14] R. Ning, W. Bu, Y. Ju and S. Duan, 'A survey of detection methods research on living-off-the-land techniques,' vol. 1, SciTePress, 18 Aug. 2023. DOI: 10.1109/icsece58870.2023.10263445. [Online]. Available: https://doi.org/10.1109/icsece58870.2023.10263445.

[15] 'Yara - the pattern matching swiss knife for malware researchers.' (), [Online]. Available: https://virustotal.github.io/yara/ (visited on 29/01/2024).

[16] M. Tsai, C. Lin, Z. He, W. Yang and C. Lei, 'Powerdp: De-obfuscating and profiling malicious powershell commands with multi-label classifiers,' *IEEE Access*, vol. 11, pp. 256–270, 1 Jan. 2023. DOI: 10.1109/access.2022.3232505. [Online]. Available: https://doi.org/10.1109/access.2022.3232505.

[17] T. Ongun, J. W. Stokes, J. B. Or *et al.*, 'Living-off-the-land command detection using active learning,' 6 Oct. 2021. DOI: 10.1145/3471621.3471858. [Online]. Available: https://doi.org/10.1145/3471621.3471858.

[18] D. 'Microsoft defender for endpoint.' (19 Jan. 2024), [Online]. Available: https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/microsoft-defender-endpoint?view=o365-worldwide (visited on 08/02/2024).

[19] T. Boroş, A. Cotaie, A. Stan, K. Vikramjeet, V. Malik and J. K. Davidson, 'Machine learning and feature engineering for detecting living off the land attacks,' 1 Jan. 2022. DOI: 10.5220/0011004500003194. [Online]. Available: https://doi.org/10.5220/0011004500003194.

[20] A. Bhardwaj, K. Kaushik, A. Alomari, A. Alsirhani, M. M. Alshahrani and S. Bharany, 'Bth: Behavior-based structured threat hunting framework to analyze and detect advanced adversaries,' *Electronics*, vol. 11, no. 19, p. 2992, 21 Sep. 2022. DOI: 10.3390/electronics11192992. [Online]. Available: https://doi.org/10.3390/electronics11192992.

[21] P. Najafi, W. Pünter, F. Cheng and C. Meinel, 'You are your friends: Detecting malware via guilt-by-association and exempt-by-reputation,' *Computers & Security*, vol. 136, p. 103 519, 1 Jan. 2024. DOI: 10.1016/j.cose.2023.103519. [Online]. Available: https://doi.org/10.1016/j.cose.2023.103519.

[22] S. Fan, Z. Liu and L. Perigo, 'Strategic monitoring for efficient detection of simultaneous apt attacks with limited resources,' *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, 1 Jan. 2023. DOI: 10.14569/ijacsa.2023.0140303. [Online]. Available: https://doi.org/10.14569/ijacsa.2023.0140303.

[23] Microsoft. 'Vssadmin delete shadows.' (3 Feb. 2023), [Online]. Available: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/vssadmin-delete-shadows (visited on 04/02/2024).

[24] 'Certutil — lolbas.' (), [Online]. Available: https://lolbas-project.github.io/lolbas/Binaries/Certutil/ (visited on 05/02/2024).