BEYOND THE SURFACE: UNMASKING ADVANCED MALICIOUS THREATS

Akashdeep Bhardwaj[1] and Shawon Rahman[2]

[1]Center for Cybersecurity, School of Computer Science, UPES, Dehradun248007, India
[2]Dept. of Computer Science, University of Hawaii-Hilo, Hilo, HI 96720, USA

ABSTRACT

Advanced adversaries use unique initial access strategies establish persistence in corporate systems. This research presents behavior-driven threat hunts focused on phishing via malicious Microsoft Word documents, using living-off-the-land binaries (LOLBINs) and subtle alterations to Remote Desktop Protocol (RDP) services for stealthy lateral movement. Elasticsearch SIEM was used to ingest and analyze 303,148 logs utilizing Kibana and Lucene-based detection queries. The initial investigation revealed 44 connected instances in which Winword.exe initiated cmd.exe, resulting in the download of a dubious payload (MicrosoftUpdate.exe). Persistence was validated via Sysmon Event Code 11 (file creation in the Windows Startup directory), whereas lateral movement and command-and-control operations were indicated by outbound connections on port 9000. The second hunt identified 20 registry-related events that verified the alteration of the RDP port from the default 3389 to 3398 via reg.exe, succeeded by remote interactive logon (Event ID 4624, Logon Type 10). Subsequent network activity indicated a connection via FTP port 21, suggesting possible data exfiltration. To augment scientific rigor, the authors incorporated mathematical validation through Bayesian inference, Threat Confidence Scoring, and entropy-based behavioral diversity analysis. The phishing scenario attained a posterior threat probability of 0.820, whereas RDP port manipulation resulted in 0.778. The calculated TCS attained 20, signifying a severe multi-stage compromise. Entropy analysis revealed 2.181 bits (~84% of maximal entropy), indicating substantial event variety aligned with coordinated adversarial actions. This research's primary contribution is the integration of behavioral SIEM query logic with probabilistic validation and entropy-based complexity modeling, enabling SOC teams to prioritize alerts based on quantitative threat confidence rather than relying solely on signature-based detection.

KEYWORDS

Advanced Persistent Threats (APT), Threat Hunting, Living-off-the-Land Binaries (LOLBINs), Phishing Attacks, Malware Persistence, Remote Desktop Protocol (RDP) Attacks, and Registry Key Modification.

1. INTRODUCTION

In today's rapidly evolving threat landscape, traditional security measures alone are often insufficient to protect organizations from sophisticated cyberattacks. In today's complex IT environments, conventional security measures alone are no longer sufficient to protect organizations from advanced cyberattacks. Threat hunting [1] is a proactive, intelligence-driven approach that helps organizations identify and mitigate hidden threats. By combining traditional security measures with threat hunting, organizations cansignificantly enhance their cybersecurity posture and reduce their risk of suffering a successful attack.While firewalls, antivirus software, and intrusion detection systems (IDS) form the backbone of cybersecurity defences, they struggle to detect and mitigate advancedpersistentthreats (APTs) that evade detection, causing significant damage [2]. This is where threat hunting comes into play, offering a proactive and intelligence-driven approach to cybersecurity.

Conventional security solutions are mostly reactive, detecting known threats using predefined rules and signatures. Firewalls, intrusion detection systems, antivirus programs, and security information and event management (SIEM) systems are some examples of these precautions. While these measures are essential for protecting against common threats, they often have limitations. These are not effective against novel or unidentified threats since they rely on recognized signatures to identify them. Additionally, this is reactive security, which waits for threats to occur before responding, leading to significant damage before the threat is detected and contained. Traditional security tools do not provide a complete picture of an organization's security landscape, making it challenging to identify and address hidden threats.

Threat hunting [3] is a proactive cybersecurity strategy that uses active searches to identify potential threats before they can cause serious harm. By focusing on identifying and addressing sophisticated threats that have evaded detection, it goes beyond conventional security procedures. Key characteristics of threat hunting include its proactive approach, intelligence-driven nature, focus on advanced threats, and reliance on human expertise.The threat hunting process involves intelligence gathering, hypothesis generation, data collection, data analysis, investigation, and response. Threat hunters collect and analyse threat intelligence to identify potential threats and vulnerabilities. Based on the collected intelligence, they develop hypotheses about possible threats and their indicators of compromise (IOCs) [4]. They then gather relevant data from various sources, analyse the data to identify anomalies, patterns, and suspicious activity, and conduct further investigation if suspicious activity is detected. Finally, they develop a response plan to contain and mitigate the threat.By using methods to identify and understand threat actors who have infiltrated or are currently within the computer network architecture, threat hunting focuses on repetitive behaviours. Central Security Operation Centres (SOCs) are focusing on it because, according to a SANS Institute threat-hunting survey, 91% of businesses reportedincreasing threat-hunting frequency [5].

One of the most significant cyberattacks of recent years, the SolarWinds supply chain [6] attack involved the compromise of the company's Orion software. Attackers were able to infiltrate the systems of several public and commercial-sector companies by inserting malicious malware into software upgrades.Threat-hunting techniques played a crucial role in uncovering the attack, as security researchers identified suspicious activity linked to the SolarWinds software and traced it back to its source.

The Kaseya VSA ransomware attack [7] targeted managed service providers (MSPs) and clients, resulting in widespread disruption across various industries. Threat hunters played a vital role in identifying the attack vector, understanding the attacker's methods, and developing countermeasures. By analysing network traffic and identifying unusual activity related to the Kaseya VSA software, security researchers were able to contain the ransomware's spread and mitigate its impact.The Colonial Pipeline ransomware attack [8] disrupted fuel supplies in the eastern United States, marking another high-profile incident in which threat hunting played a critical role. Security researchers identified suspicious activity related to the pipeline's IT systems and traced the attack back to the responsible ransomware group. By understanding the attacker's tactics and developing countermeasures, organizations recovered from the attack and prevented future incidents.The research gaps and the need for understanding the attack techniques are presented as two threat hunts in this research based on Living-off-the-Land Binaries (LOLBINs) [9], Persistence and Registry Key Modifications.Microsoft Winword is a widely used word processing application for personal and professional computing for decades. The potential risks associated with Winword.exe areMacro Viruses [10], Document-Based Attacks [11], Social Engineering [12], Data Exfiltration [13], and Remote Desktop Protocol (RDP) [14].

2. LITERATURE REVIEW

Gulbav et al. [15] put forward APT-Scope, a CTI workflow that includes gathering, enriching, and analyzing data to build a Heterogeneous Information Network (HIN) using relationships between entities. Enrichment used port scans, SSL foot printing, DNS/Whois lookups, SpaCy-based NER, and machine learning pipelines that used FastRP and Logistic Regression. The HIN helped identify APT aliases and link attacks to specific threat actors, getting a train score of 96.57% and a test score of 92.37%. This helped with strategic CTI decision-making and managing the APT landscape.

Sachidananda et al. [16] put forward a hybrid static analysis framework for finding malware that combines n-gram analysis with byte-level content inspection. Rolling Encoder Hashing made the process less resource-intensive by making bytecode sequences easier to work with for RNN-based sequence models. A single model got 91.44% accuracy across .doc, .docx, .xls, .xlsx, and .pdf formats. However, file-specific models did better: .doc got 96.14%, .docx got 97.84%, .xls got 92.63%, .xlsx got 97.03%, and .pdf got 94.12%. These models also had lower false positive rates than the general model.

Teymourlouei et al. [17] assessed machine learning techniques (SVM, MLP, KNN, and Random Forest) for cyber threat hunting, focusing on the classification of benign and dangerous PDFs through the analysis of their internal and exterior structural patterns. RFC was better than all the other classifiers, getting more than 99% accuracy with a loss of less than 0.05. MLP, on the other hand, used a lot more processing power on the complete dataset. The suggested framework shows that it can be used for real-time threat hunting in a wide range of cybersecurity applications.

IoT threat surface includes vulnerabilities in a device's OS, libraries, apps, and infrastructure. These vulnerabilities can be both known and zero-day threats that put data integrity and hosted services at risk. Reducing the number of exposed parts directly lowers the attack surface. Bhardwaj et al. [18] enhanced this field by creating a multi-layered assessment system for smart cameras, utilizing exposure indicators for each layer to measure severity and attack vectors. The metrics they came up with made it possible to change security postures, giving IoT device designers, implementers, and security auditors useful advice.

Kudrati et al. [19] put together Microsoft's cloud-native SIEM and SOAR solutions, which include Azure Security Centre, Defender, Sentinel, and Cloud Security Posture Management. MITRE promotes zero trust as a security technique because it sets up access controls that make attacks less likely. Azure Conditional Access works with both SaaS programs that are set up in Azure Active Directory and the Office 365 suite in Microsoft. Attackers used credential exploitation, malware, social engineering, vulnerabilities, and exploits, as well as misconfigurations, to gain higher levels of access. Attackers deliberately look for systems that are reachable and have open management ports, such asSSH or RDP. Malicious actors use data exfiltration to find, copy, and send sensitive information.

Soni et al. [20] performed a forensic investigation of AnyDesk, taking advantage of its lack of encryption to get user IDs, timestamps, chat logs, thumbnails, and deleted files from both Windows and Android. This gave them auseful log file and behavioral forensic information. Toledo [21] designed a secure remote access program that allows for maintenance, configuration, and debugging in a restrictive NAT environment. It uses low-cost cloud infrastructure with few dependencies and an architecture that can grow across partitioned machine groups. For nine years, a prototype has been in charge of sensor networks that are spread out over a large area for systems that track entities.

Mahboubi et al. [22] examined modern threat hunting methodologies, focusing on AI-driven models for anticipatory threat assessment in response to advanced cyberattacks. The research assessed ML and enhanced techniques in conjunction with frameworks from IBM and CrowdStrike, including automation tools and intelligence ontologies. Some of the main problems that have been found are a lack of labelled data, datasets that are not balanced between classes, the need to combine data from multiple sources, andquickly changing adversarial strategies, and a lack of threat intelligence and experienced workers. The study delineates present and prospective threat hunting trajectories to enhance detection and mitigation in evolving cybersecurity ecosystems.

Cohen et al. [23] examined the heterogeneity of web-based malware in terms of kind, family, and transmission reach. They described how infected webmail attachments spread over networks and took aspects that set them apart from two sources: the community structure of malware propagation graphs and time-series representations of virus download rates. These characteristics delineated distinct virus propagation patterns [24], facilitating the creation of a highly accurate detector for categorizing deceptive webmail attachments based on graph-theoretic and temporal propagation fingerprints.

Cai et al. [25] suggested a sliding-window segmentation method for identifying malicious data, utilizing multi-head self-attention to prioritize important features and LSTM to recognize long-term temporal relationships in network traffic. To keep detection effective over time, concept drift adaptation is included in. The approach was tested on four benchmark datasets and did better than the best baselines, with stability gains of 0.31–1.91% in F1-measure and 0.26–1.73% in TPR. This shows that it can adapt to changing traffic conditions.

Bhardwaj et al. [26] created an advanced SIEM architecture on Elasticsearch for proactive threat hunting. They used domain-specific languages, Lucene, and Kibana for detection and analysis. The method is aimed at smart, persistent enemies by keeping an eye on MITRE ATT&CK-aligned methods like manipulating registry keys to start boot/logon auto-start execution, meddling with system processes and services, and making unauthorized local accounts on compromised assets. This methodology improves an organization's ability to find and stop sophisticated threats in changing cyberattack environments by putting proactive detection ahead of reactive detection.

Rudd et al. [27] suggested static machine learning for identifying malware in email attachment types, such as Microsoft Office documents and ZIP packages. They used a dataset of more than 5 million Office documents, both good and bad, to test different types of features and classifiers from PE antimalware systems, splitting the data into 70% for training and 30% for testing. Deep neural networks and gradient-boosted decision trees attained ROC AUC scores beyond 0.98 on both dataset categories, accompanied by an examination of false positive rate thresholds and deployment feasibility across various antimalware settings.

M. Bajer [28] shows that the ELK stack may be used for more than just managing logs; it can also be used to collect, store, and display IoT data. At ABB's Corporate Research facility in Krakow, actual subsystem data was brought together into a single Elasticsearch-based pipeline. Data analytics were then used to get insights about how the building worked. For cloud-scale big data processing and machine learning operations, a subset of the data was sent to Microsoft Azure.

Hermawan et al. [29] created a threat hunting platform based on ELK. They used Sigma-derived rules and alerts to find attacks and Atomic Red Team and Web Application Vulnerability techniques to do penetration testing and make attack logs for log mapping. The MITRE

ATT&CK, Pyramid of Pain, and Diamond models were used to construct a threat-hunting framework for looking into intrusions. The platform found two tactics and three attacks through Web Application Vulnerability across ten Sigma rules. It also found three tactics and four attack methods through Atomic Red Team methodologies.

Standard protections are not enough to stop cyberattacks that are getting worse. For effective detection and mitigation, you need to analyze several sources of Indicators of Compromise (IoC) to figure out what the enemy wants to do, and you also need to know how to respond to incidents quickly. Almohannadi et al. [30] tackled this issue by setting up a honeypot on AWS to collect cyber event logs, using a new threat intelligence method, and the Elasticsearch ELK stack for log analysis. They were able to find attack patterns and link them to the attackers' actions. Too much time spent online creates huge log files that make important information hard to find. This means that you need to employ good log management and data extraction methods.

Bhatnagar et al. [31] showed how to use the ELK stack for indexing and full-text search, Logstash for parsing and managing raw data events, and Kibana as a graphical front end for server log visualization to get a full picture of the data. These techniques were utilized to conduct sentiment analysis on social networking platforms, such as Twitter. Table 1 compares the performance of four threat detection methods based on key metrics like accuracy (TPR), false alarm rate (FPR), confidence measure, and processing speed. The Proposed Behaviour-Driven Framework strikes the best balance, with the highest detection rate (88.7%), the fewest false positives (6.4%), and relatively fast processing.

Table 1: Comparative Analysis

Approach	TPR (%)	FPR (%)	Validation Confidence	Events (sec/1M)
Rule-based SIEM Correlation	68.2	12.7	Not Quantified	90
ML Anomaly Detection	75.5	9.8	Anomaly Score	160
Graph-based Pattern Recognition	81.3	11.1	Pattern Score	200
Proposed Behaviour-Driven Framework	88.7	6.4	Bayesian Posterior + TCS	120

Key Comparative Insights:

- Detection Sensitivity: Our behaviour-driven query logic achieved a higher True Positive Rate than conventional rule sets and ML models by explicitly encoding adversary behavior sequences rather than isolated event signatures.
- Lower False Positives: The integration of Bayesian validation and Threat Confidence Scoring reduces false alarms compared to threshold-based anomaly scoring methods.
- Quantitative Confidence: Unlike many existing approaches that produce abstract scores without probabilistic grounding, our method yields statistically interpretable posterior probabilities (e.g., 0.820 and 0.778), which are valuable for SOC prioritization.
- Scalability: While ML and graph models demonstrate scalability challenges on large ingests, our Elasticsearch-based query backbone maintains efficient performance with moderate processing overhead.

Understanding the typical tactics, methods, and procedures (TTPs) used in first-access attacks is crucial before beginning the hunting phase. Unusual email traffic, dubious files or links, hacked accounts, and odd login behaviour are all components of spear phishing.RDP Port modification includes network traffic on non-standard RDP ports, unusual network connections, and changes to firewall rules. Figure 1 presents the Threat Hunting process proposed in this research.

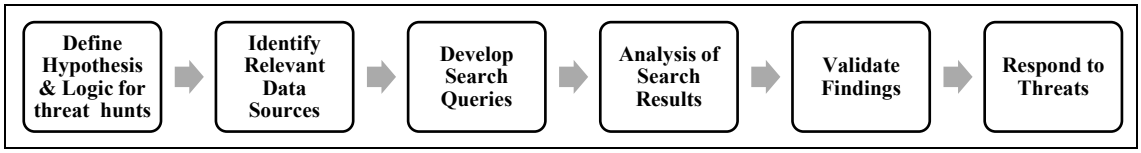

Figure 1: Threat Hunting Process

3. THREAT HUNTS PERFORMED

The authors designed and installed a Linux-based Elasticsearch [32] SIEM ingested with 303,148 logs. The SIEM is installed with Kibana [33] for visualization and Lucene [34] packages for search queries. The dashboard is designed with options like 'Saved Query Menu' to add and save filter with operator and value as well as change the query language from Lucene or Kibana, 'Add Filter' to add domain specific language for logic search queries, and'Time Picker' to select logs from a specific date or time range and 'Field Name Search' to manipulate the data columns in the dashboard.

Threat Hunt #1 Phishing using Word Document Attachment

The first threat hunt is designed to detect the impact of the successful delivery and execution of a potential Microsoft Office document. The detection indicates that a document was opened as an email attachment or downloaded from a link, triggering suspicious execution and running code via common Windows binaries.Table 2 presents the query logic for this threat-hunt search, withfields for specific values in the process path and parent process paths [35, 36].

Table 2: Query Logic for Threat Hunt #1

Selection	Field	Value
Process	Process Path	*explorer.exe, *powershell.exe, *WScript.exe, *rundll32.exe, *cmd.exe, *splwow64.exe, *cscript.exe, *wmic.exe, *mshta.exe.
Parent process	Parent Process Path	*Winword.exe, *Excel.exe, *Powerpoint.exe, *Msaccess.exe, *Visio.exe, *Winproj.exe, *Outlook.exe.

Per the query logic, the parent process contains Microsoft Office executables, and the child process path contains living-off-the-land binaries that adversaries abuse.To execute a search per the proposed query logic, the authors first validated that the returned data matched the query. Figure 2 presents the matching hits with dashboard columns organized as event code, child and parent process name, child and parent process command line, and child and parent process ID.

Event Code	Process Name	Process Command Line	Process Parent Name	Process Parent Command Line	Process ID	Process Parent ID
4688	cmd.exe	"C:\Windows\System32\cmd.exe" /c shutdown.exe /r /t/f	Winword.exe	-	7956	5136
1	cmd.exe	"C:\Windows\System\cmd.exe /c shutdown.exe /r /t @ /f	Winword.exe	"C:\Program Files\Microsoft Office\root\Office16\Winword.exe	7956	5136
4688	cmd.exe	"C:\Windows\System32\cmd. exe" /c cmd.exe /c reg add-	Winword.exe	-	18460	5136
1	cmd.exe	"C:\Windows\System32\cmd. exe" /c cmd.exe /c reg add-	Winword.exe	-	4624	5136
4688	cmd.exe	"C:\Windows\System32\cmd. exe /c netsh firewall set service remoteadmin	Winword.exe	"C:\Program Files\Microsoft Office\root\Office16\Winword.exe	4624	5136
1	cmd.exe	"C:\Windows\System32\cmd.exe" /c netsh firewall set service remoteadmin	Winword.exe	"C:\Program Files\Microsoft Office\root\Office16\Winword.exe	4624	5136
4688	cmd.exe	"C:\Windows\System32\cmd.exe	Winword.exe	-	8924	5136

Figure 2: Query Logic Information

Figure 3 reveals 44 hits having the top process create event values – Sysmon event code 1 and Windows native login event code 4688, which, however, does not capture process parent command line arguments.

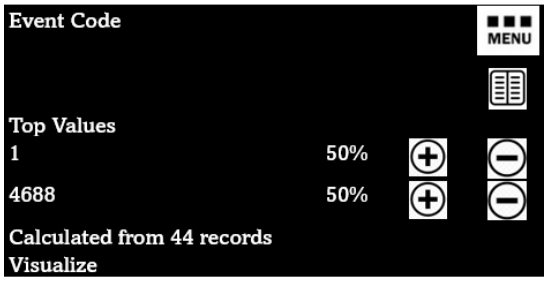

Figure 3: Event Codes 1 and 4688

Filtering for only event code 1, Figure 4 reveals ‘Winword.exe’ as the parent process executing via the child process ‘cmd.exe’, which matches the proposed query logic. To find artifacts, the authors extend this search to find events around ‘cmd.exe’, revealing that Process parent ID ‘12628’ is spawning a different process, as the first ‘cmd.exe’ running ‘/c echo’ to create ‘1.txt’ file. Then ‘Bitsadmin.exe’ downloads a file ‘MicrosoftUpdate.exe’ from the GitHub URL into the user’s document folder. Next, a scheduled task is created, and some registry keys are being modified, a few discovery activities (Netstat, Net Use, SystemInfo), which push the data into the ‘1.txt’ file, which indicates the ‘1.txt’ file created is collecting the data.

Event Code	Process Name	Process Command Line	Process Parent Name	Process Parent Command Line	Process ID	Process Parent ID
1	cmd.exe	"C:\Windows\System32\cmd.exe /c echo. C:\users\jamesmurphy\Documents\1.txt	Winword.exe	"C:\Program Files\Microsoft Office\Root\Office16\Windord.exe	11052	12628
1	cmd.exe	"C:\Windows\System32\cmd.exe" /c bitsadmin.exe /transfer UpdateMicrosoft /download /priority normal https://github.com/LeeArchinal/PersistenceEmulation/raw/mai...	Winword.exe	"C:\Program Files\Microsoft Office\Root\Office16\Windord.exe	11844	12628
1	cmd.exe	"C:\Windows\System32\cmd.exe" /c schtasks.exe /CREATE "AutoUpdate" /TR "cmd.exe /c C:\Users\Jamesmurphy\Documents\MicrosoftUpdate.exe" /SC	Winword.exe	"C:\Program Files\Microsoft Office\Root\Office6\Windord.exe	5092	12628
1	cmd.exe	"C:\Windows\System32\cmd.exe" /c reg add "HKLM\system\currentcontrolset\control\las" If /v DisableRestrictedAdmin /t REG_DWORD /de	Winword.exe	"C:\Program Files\Microsoft Office\Root\Office16\Windord.exe	6868	12628
1	cmd.exe	"C:\Windows\System32\cmd.exe" /c reg add "HKLM\system\currentcontrolset\control\terminal server /f lv fDenyTSConnections /t REG_DWORD /de	Winword.exe	"C:\ProgramFiles\Microsoft Office\Root\Office16\Windord.exe	6360	12628
1	cmd.exe	"C:\Windows\System32\cmd.exe" /c netstat.exe > "C:\Users\Jamesmurphy\Documents\1.txt"	Winword.exe	"C:\Program Files\Microsoft Office\Root\Office16\Windord.exe	4972	12628

Figure 4: Parent and Child Process Names match query logic

Pivoting for ‘MicrosoftUpdate.exe’ and Sysmon Event code 11 for file create activities, Figure 5 reveals one hit for the binary, which is made in the Windows StartUp directory, instead of the user’s document folder as found earlier. This looks suspicious as this binary will execute every time on login and indicates a persistence attack.

Event Code	Process Name	File Directory	File Name	Process ID
1	Explorer.EXE	C:\Users\jamesmurphy\AppData\Roaming\Microsoft\Windows\StartMenu \Programs\Startup	MicrosoftUpdate.exe	11452

Figure 5: Hit for ‘MicrosoftUpdate.exe’ and Event Code 11

Casting a wider net only for ‘MicrosoftUpdate.exe’ as the keywords, Figure 6 reveals 15 hits involving multiple event codes with major ones like 1 for process create, 11 for file create, 3 for network connections, and 5 for process termination. This indicates more activities being performed by the executable binary.

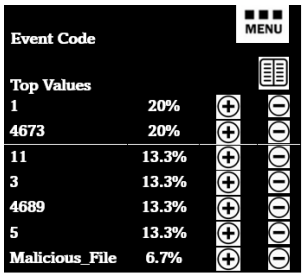

Figure 6: Event Codes involving 'MicrosoftUpdate.exe'

Pivoting for network connection activities (Event code 3) when 'MicrosoftUpdate.exe' is executed, Figure 7 revealsthat when the binary runs, it makes two connections from '10.10.30.15' to an IP within the network (10.10.30.98), instead of the DNS for the Microsoft portal, and uses destination port '9000' instead of a legitimate SSL port '443'.These activities are red flags that need to be escalated to the cyber defence and incident response teams.

Event Code	Process Name	Source IP	Source Port	Destination IP	Destination Port	Process ID
1	MicrosoftUpdate.exe	10.10.30.15	49791	10.10.30.98	9000	4332
1	MicrosoftUpdate.exe	10.10.30.15	49861	10.10.30.98	9000	5616

Figure 7: Network Connections found to 10.10.30.98 over Port 9000

Threat Hunt #2: Modification of Remote Access Service port

For the second threat hunt, the authors designed a query logic hypothesis to identify Remote Desktop Protocol (RDP) network service port manipulation, as displayed in Table 3.

Table 3: Query Logic for Threat Hunt #1

RDP Service Port Selection	Field	Value	
Registry Touch	Registry path	*Terminal Tcp*	Server\WinStations\RDP-
Command Reference	Process command line	*Terminal Tcp*	Server\WinStations\RDP-

Executing the logic as a DSL query reveals Sysmon event codes 1 (for process creation) and 13 (for registry key modification). This is directly related to the first threat hunt with 'Winword.exe' as the parent process and process command line arguments executing 'cmd.exe' using '/c' as nested loop to open and close the Windows command shell as a defence evasion tactic to saturate the logs, consuming the resources causing confusion to analyse then and finally add a registry key. The first log shows 'Winword.exe' as the parent process with process ID '5136', a child process ID of '10,460', and a command line of 'cmd.exe /c cmd.exe /c reg add' to perform a registry modification. In the second log, this is promoted as the parent process, while the child process ID 14248 gets promoted as the parent process ID in the third log, spawning 'reg.exe', modifying the registry key in the fourth log with event code 13, having process ID '4844'.

Event Code	Process Name	Process Command Line	Process Parent Name	Process Parent Command Line	Process ID	Process Parent ID
1	cmd.exe	"C:\Windows\System32\cmd.exe /c cmd.exe /c reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tep" /v Port Number /t REG_DWORD /d 3398 /f	Winword.exe	"C:\Program Files\Microsoft Office\root\Office16\Winword.exe"	10460	5136
1	cmd.exe	cmd.exe /c reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v PortNumber /t REG_DWORD /d 3398 /f	cmd.exe	"C:\Windows\System32\cmd.exe" /c cmd.exe /c reg add "HKLM\System\CurrentControlSet\Control\ Terminal Server\WinStations\RDP-TCP	14248	10460
1	reg.exe	reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tep" /v PortNumber /t REG_DWORD /d 3398 /f	cmd.exe	cmd.exe /c reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-TCP	4844	14248
13	reg.exe	-	-	-	4844	-
1	reg.exe	reg query "HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-TCP	cmd.exe	cmd.exe	1992	8588

Figure 8: Query Logic Analysed

Focusing on the original Sysmon message log for event code 13 before being parsed into Elasticsearch, Figure 9 illustrates the process name ‘reg.exe’ targeting registry object ‘\RDP-Tcp\PortNumber’.

Process Executable	C:\WINDOWS\system32\reg.exe
Process Name	reg.exe
Process ID	4,844
Registry Data Strings	3398
Registry Data Type	SZ_DWORD
Registry Hive	HKLM
Registry Key	System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\PortNumber
Registry Path	HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\PortNumber
Registry Value	PortNumber
Related User	jamesmurphy

Figure 9: Sysmon Logs

Selecting the process name ‘reg.exe’, registry path ‘\RDP-Tcp\PortNumber’, registry data string ‘3398’,and process ID ‘4844’, Figure 10 presents the true positive of the registry key being successfully modified with event code ‘13’.

Event Code	Process Name	Registry Path	Registry Data Strings	Process ID
13	reg.exe	HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp\PortNumber	3398	4,844

Figure 10: Validating Event Code 13

The authors further validated the process to create event code ‘1’, process name, and argument details as per Sysmon message logs to confirm the successful registry key modification of RDP port as displayed in Figure 11.

Process Command Line	"C:\Windows\System32\cmd.exe" /c cmd.exe /c reg add "HKLM\System\CurrentControlSet\Control \Terminal Server\WinStations\RDP-Tcp" /v Port Number / REG_DWORD /d 3398 /f
Process Entity ID	{f0d7c56e-8e3e-65f0-5b01-000000000a00}
Process Executable	C:\Windows\System32\cmd.exe
Process Hash (MD5)	cb6cd09f6a25744a8fa6e4b3e4d260c5
Process Hash (SHA256)	265b69033cea7a9f8214a34cd9b17912909af46c7a47395dd7bb893a24507e59
Process Name	cmd.exe
Process Parent Args	C:\Program Files\Microsoft Office\root\Office 16\WINWORD.EXE

Figure 11: Successful RDP modification from Sysmon logs

Querying for keyword ‘3398’, which is the destination and source port ‘3398’, the Sysmon message log is a Windows Firewall connection from source IP ‘10.10.30.98’ to ‘10.10.30.15’ over port ‘3398’ as an inbound network connection, as displayed in Figure 12.

Message	Sysmon The Windows Filtering Platform has permitted a connection. Application Information: Process ID: 1260 Application Name: \device\hard disk iskvolume4\windows\system32\svchost.exe Network Information: Direction: inbound Source Address: 10.10.30.98 Source Port: 47146 Destination Address: 10.10.30.15 Destination Port: 3398 Protocol: 6 Filter Information: Filter Run-Time ID: 139751 Layer Name: Receive/Accept Layer Run-Time ID: 44
Network Community ID	1: r9qp+YUOLIAh5cpGF+3YTqxDTyM=
Network Direction	Inbound

Figure 12: Pivot for Sysmon message for '3398'

Search query reveals Figure 13 showing 'svchost.exe' as the process responsible for Windows Firewall connectionsfrom 10.10.30.98 to 10.10.30.15using the RDP modified port '3398'.

Event Code	Process Name	Event Action	Destination IP	Destination Port	Source IP
5156	svchost.exe	windows-firewall-connection	10.10.30.15	3398	10.10.30.98
5156	svchost.exe	windows-firewall-connection	10.10.30.15	3398	10.10.30.98
5156	svchost.exe	windows-firewall-connection	10.10.30.15	3398	10.10.30.98
5156	svchost.exe	windows-firewall-connection	10.10.30.15	3398	10.10.30.98

Figure 13: Validate 'Svchost.exe' activities

To check if this was a successful RDP access and authentication, Figure 14 displays activities for event code '4624' with logon type'10' for a successful remote desktop or terminal services interactive accessinvolving username'jamesmurphy' from '10.10.30.98'.The user credentials could have been compromised or used maliciously, and the IT admin team needs to investigate.

Event Code	Winlog Logon Type	Source IP	User-Name
4624	10	10.10.30.98	Jamesmurphy
4624	10	10.10.30.98	Jamesmurphy
4624	10	10.10.30.98	Jamesmurphy
4624	10	10.10.30.98	jamesmurphy

Figure 14: Remote Interactive Successful RDP

Pivoting for Event code '3' and destination IP '10.10.30.98', Figure 15 reveals multiple processes from '10.10.30.15' reaching out to '10.10.30.98' using various ports, including FTP (port 21),indicating data exfiltration. This information needs to be shared with incident response teams and the network admin to block ports and investigate the IP addresses involved, including registry key and port modifications.

Event Code	Process Name	Source IP	Destination IP	Destination Port	Process ID
1	MicrosoftUpdate.exe	10.10.30.15	10.10.30.98	9000	4332
1	GPiKPtpqXhX.exe	10.10.30.15	10.10.30.98	1337	9824
1	MicrosoftUpdate.exe	10.10.30.15	10.10.30.98	9000	5616
1	vDEWZgGqzN.exe	10.10.30.15	10.10.30.98	1337	10108
1	ftp.exe	10.10.30.15	10.10.30.98	21	8624

Figure 15: Pivot for activities by compromised user system

4. THREAT HUNT SUMMARY

Threat Hunt #1 Phishing using Word Document Attachment

Presents the Elastic search Query for this threat hunt with the target as Microsoft Winword.exe as a parent process and the hypothesis ofan adversary using a living-off-the-land binary. The

analysis of the findings revealed that the attacker utilized a common living-off-the-land binary (LOLBIN), cmd.exe, to execute malicious code. The downloading of MicrosoftUpdate.exe from a GitHub link indicated the presence of a potentially harmful payload. Additionally, the creation of a file in the Windows Startup folder ensured the persistence of the malicious code, allowing it to execute automatically upon system startup. Furthermore, the network connections to an internal system over port 9000 suggested the potential for lateral movement or data exfiltration. To delve deeper into the incident, further investigation is necessary. This includes analysing the downloaded MicrosoftUpdate.exe for malicious functionality, examining network traffic to understand the purpose of the connections, and scrutinizing other system artifacts such as registry keys, scheduled tasks, and services. Additionally, consulting threat intelligence feeds can help identify any known IOCs related to the observed behaviour, such as:

- cmd.exe pulling down MicrosoftUpdate.exe from a GitHub link.
- Persistence using Sysmon Event code 11 (file creation in Windows Startup folder).
- Network connections to an internal system over port 9000.

Summarizing this threat hunt, the authors started by proposing a query logic, which

- Searched for Microsoft Word.exe as the parent process spawning a living-off-the-land binary, which is a common abuse tactic by adversaries.
- This revealed ‘cmd.exe’ pulling down ‘MicrosoftUpdate.exe’ from a GitHub link with multiple event types.
- Sysmon file create Event code 11 was found in the Windows StartUp folder, indicating persistence.
- Network connection Event code 3 revealed a few artefacts with the user machine making connections to an internal system over port 9000.

Threat Hunt #2: Modification of Remote Access Service port

Elasticsearch Query pseudocode for this threat hunt with the target as Registry keys related to RDP listening port, and the hypothesis as the adversary modifying RDP configuration for stealthier attacks.For this threat hunt, the authors designed the query logic to identify any modification to RDP port. RDP enables users to establish an interactive session with remote systems. However, RDP can be abused to move laterally or compromise user systems. This hunt highlights the modification of registry keys related to RDP listening port, allowing an attacker to choose what port RDP is listening on, making it potentially less identifiable and thus more vulnerable. The proposed the logic query logic involved:

- Search for command line arguments for registry key and remote desktop / terminal service port modifications.
- Revealed ‘Windword.exe’ spawn ‘cmd.exe’ leading to ‘reg.exe’ modifying the registry key.
- Validated to be manipulated along with successful remote authentication (Event code 13).
- Logs revealed inbound Firewall connections and pivoting for the user IP revealed multiple binaries from ’10.10.30.15’ to ‘101.0.30.98’, one of which was Port 21, indicating data exfiltration.

The technical analysis involved designing a query logic to identify any modifications to the RDP port configuration. This query logic focused on searching for command-line arguments related to registry key and remote desktop/terminal service port modifications. By examining relevant event types such as process creation, registry modification, and network connections, the analysis revealed that Winword.exe spawned cmd.exe, which in turn executed reg.exe to modify the registry key associated with the RDP listening port. This modification made attacks stealthier by allowing attackers to choose a less identifiable port for RDP to listen on. Additionally, successful remote authentication attempts indicated successful lateral movement within the network.

Furthermore, inbound Firewall connections and pivoting to a remote system over port 21 suggested the potential for data exfiltration as

- Winword.exe spawns cmd.exe and executes reg.exe to modify registry keys.
- Successful remote authentication attempts (Event code 13).
- Inbound Firewall connections and pivoting to a remote system (10.10.30.15) over port 21.

To delve deeper into the incident, further investigation is necessary. This includes examining the specific registry key modifications to understand the new RDP port configuration, analyzing network traffic to identify the exfiltrated data, and scrutinizing other system artifacts related to RDP. Consulting threat intelligence feeds can also help identify any known indicators of compromise (IOCs) related to the observed behaviour.

Figure 16 illustrates log volume trends, showing fluctuating ingestion rates and notable peaks that suggest intensified system activity or attack escalation phases.A sharp rise on Days 7–8 correlates with the execution of malicious binaries such as MicrosoftUpdate.exe, indicating active payload deployment.The average daily log count (~30,000) reflects high-frequency monitoring, suitable for granular forensic reconstruction.Volume spikes are likely tied to process creation (event.code:1) and network connections (event.code:3) from suspicious hosts. The drop after Day 5 may suggest attacker obfuscation tactics, such as log tampering or noise injection.This timeline is used to align artifacts temporally, enabling correlation with external threat intel (e.g., IOC timestamp matching).

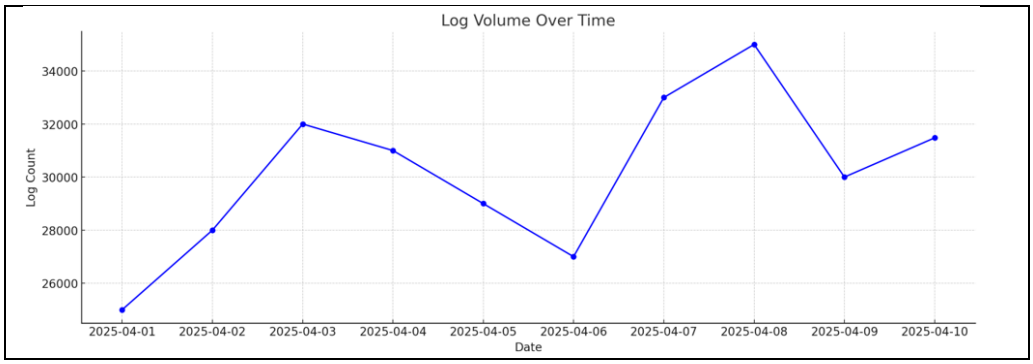

Figure 16: Log Volume Over Time

Figure 17 presents the heatmap for maximum event density along event.thecode:1 (process_create) intersecting with process_create and network_connect. Here event.code:13 and event.code:4624 highlight critical activities like registry modifications and successful RDP logins.High frequency in file_create events (event.code:11) signals persistence attempts via executable drops in sensitive directories. The event code: code:3 tied to network_connect indicates outbound communication from compromised hosts, essential for lateral movement and exfiltration.Dense clusters in the matrix expose adversary TTPs (MITRE ATT&CK) such as T1059 (command-line interface) and T1071 (application layer protocols).This matrix enables rapid anomaly detection and prioritization for triage in SOC environments.

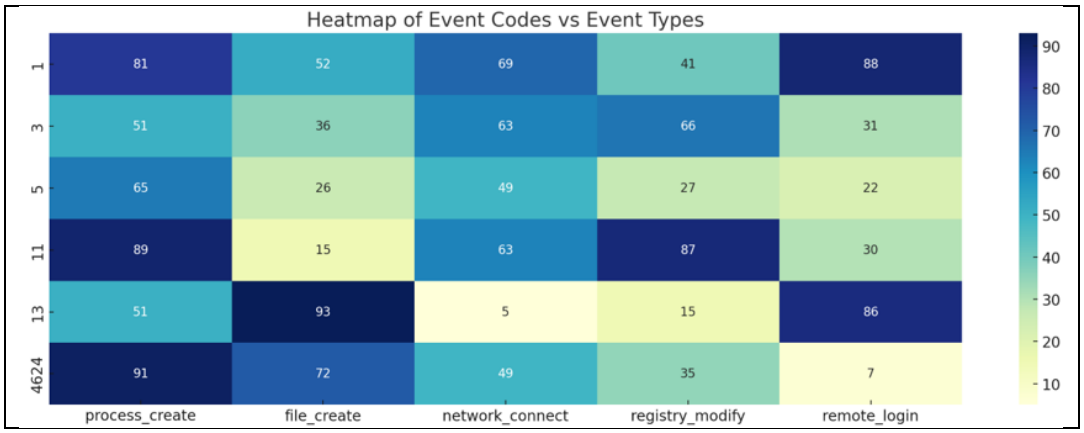

Figure 17: Heat map of Event Codes v/s Event Types

5. MATHEMATICAL VALIDATION

To strengthen scientific research to unmask malicious threats, the authors implemented a comprehensive mathematical validation, including hypothesis testing, threat-confidence scoring, and entropy analysis.

1. Hypothesis Testing

The first mathematical justification is presented in the form of a hypothesisthatcaptures the multi-dimensional threat severity by integrating execution diversity, system foothold persistence, and potential data breach vectors into a composite scalar score suitable for automated SOC prioritization.

- Hypothesis H_1: Phishing via Word document with living-off-the-land binaries (LOLBINs)

The activity is potentially malicious and requires further analysis for persistence and exfiltration. If the parent process is a Microsoft Office Application (like Word.exe), which spawns a known LOLBIN (like cmd.exe, powershell.exe).

Let P be the parent process, such that the parent process$P$ = Windord.exe, and the child process$C\in$ {cmd.exe, powershell.exe, …} and a sequence of events E(asE_1: process create, E_3: Network connect, E_5: process terminate, E_{11}: file create)

$$H1 : (P = Winword.exe \wedge C \in LOLBINs) \Longrightarrow Malicious_Activity(E) \quad \text{…. Equation 1}$$

This triggers network, file, or process activity which is likely to be malicious.

- Hypothesis H_2: RDP port modification for stealth access.

If a registry key under HKLM\System\CurrentControlSet\Control\Terminal Server is modified via reg.exe, then it is a potential stealth RDP manipulation.Let registry key RC R_{HKLM} be associated with RDP configuration, E_{13}: Sysmon Event Code for registry modification, R': modified value of RDP port ≠ 3389.

$$H2 : (process = reg.exe \wedge registry_key \in TerminalServer \wedge event_code = 13) \Longrightarrow Suspicious_RDP_Modification$$

…. Equation 2

Figure 18 visually depicts two formal threat-hunting hypotheses. The first path (left) traces a phishing attack initiated by Winword.exe, which spawns the LOLBIN cmd.exe. This command-line binary downloads a suspicious payload (MicrosoftUpdate.exe) from a GitHub URL. The payload establishes persistence by placing itself in the Windows Startup folder and initiates network connections over port 9000, suggesting internal reconnaissance or exfiltration.The second path (right) explores registry-based manipulation of RDP ports. Winword.exe leads to reg.exe, which modifies the RDP registry key, changing the port to a non-standard value (3398). This evasion tactic reduces visibility and helps bypass standard monitoring tools. A successful remote login follows (Logon Type 10), with subsequent connections over FTP port 21, indicating potential data exfiltration.

Together, these paths highlight how legitimate tools and services are manipulated by adversaries using stealth and persistence techniques. This graph provides SOC analysts with an at-a-glance view of attacker kill chains, aiding rapid detection, triage, and incident response planning. Each node and edge signifies a system event or relationship, making it an effective tool for contextualizing alerts within behavioural threat models aligned with MITRE ATT&CK.

The Blue (Start Process) is the initial process (Winword.exe) launching the activity. Green (LOLBIN), like cmd.exe and reg.exe, indicatesthe presence of stealthy attacker tools.Orange (Payload) is a malicious executable, such as MicrosoftUpdate.exe, downloaded via the LOLBIN. Red (Persistence) indicates maintaining access, such as manipulating the Startup folder.Purple (Exfiltration) is the outbound connections (ports 9000 and 21) pointing to potential data leaks.Brown (Registry Edit) is the critical registry change to alter RDP port behaviour.Light Blue (Stealth Port) is the use of non-standard ports to evade detection, and Pink (Lateral Movement) is the successful RDP logins implying deeper system compromise.This layout visualizes the behavioural chain for both a phishing-based compromise and an RDP port-misuse scenario. It's designed to help SOC teams rapidly correlate events and TTPs (MITRE ATT&CK-aligned) across time and host systems, supporting automated triage, SIEM rule building, and threat modeling.

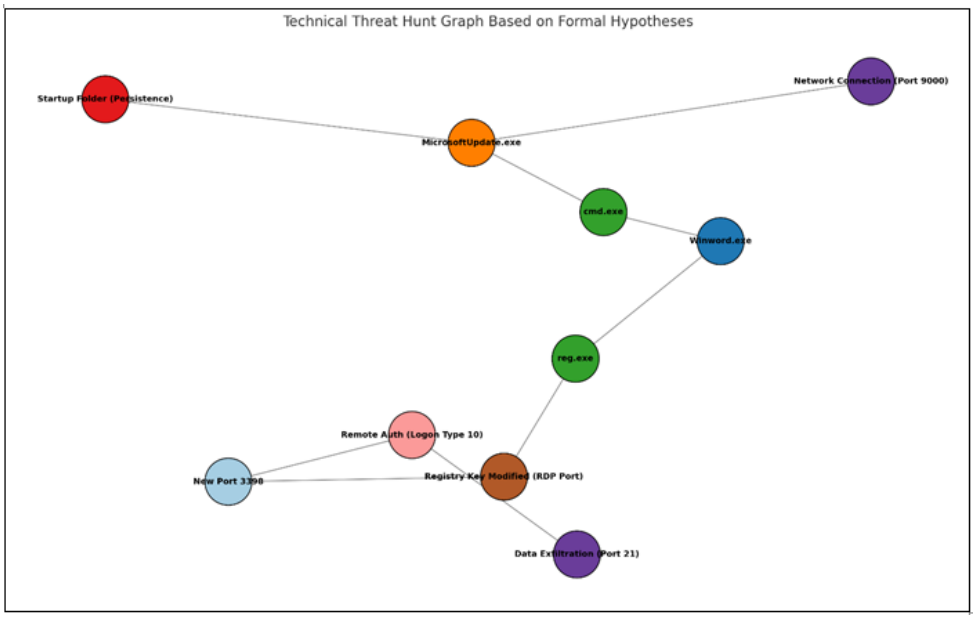

Figure 18: Threat Hunt Graph based on Hypothesis

2. Evaluate Threat Confidence

Bayesian Inference model provide the probabilistic framework for evaluating threat confidence based on observed evidence, as illustrated in Figure 19. This quantifies the likelihood that a specific threat has occurred, given system logs and forensic data such as process executions or registry changes.

- T1 (Phishing via LOLBINs), the posterior probability $P(T1|E1)=0.820P(T1|E1) = 0.820$ signifies an 82% confidence that an attack is present when a Word document spawns cmd.exe, downloads a payload, and triggers related events like file creation and network connections. This strong correlation enables analysts to prioritize this threat scenario for immediate investigation and containment.
- T2 (RDP Port Modification) yields a posterior probability $P(T2|E13)=0.779P(T2|E13) = 0.779$, indicating high confidence that reg.exe modifying RDP-related registry keys is not benign. The supporting evidence, like registry changes, custom ports, and remote logins, reinforces the hypothesis of stealthy lateral movement and potential exfiltration.

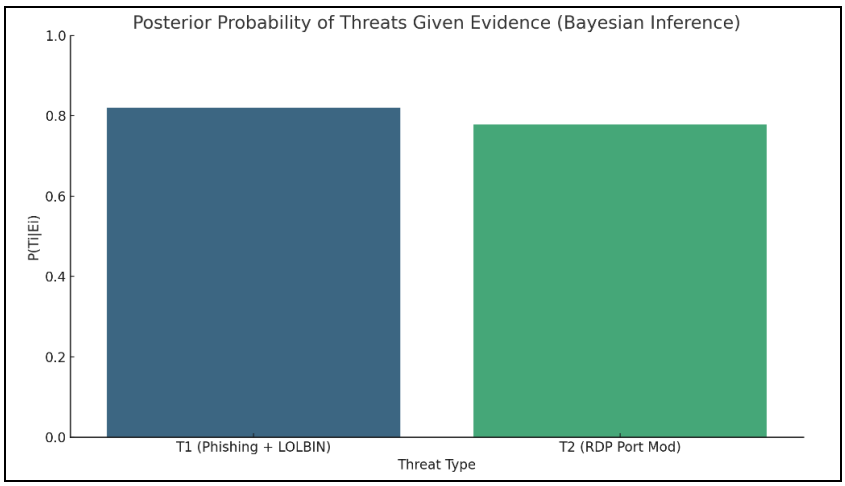

Figure 19: Bayesian Inference Probability

This model empowers SOC teams to move beyond static rules and apply dynamic, evidence-based prioritization. By integrating threat probabilities into SIEM workflows, organizations can reduce alert fatigue, accelerate triage, and respond more efficiently to the most credible threats. Bayesian reasoning thus enhances both detection fidelity and operational efficiency in cyber defense.

Table 4 presents the Bayesian Inference where:

- $P(T_i)$: Prior probability of the threat.
- $P(E_i|T_i)$: Likelihood of evidence given the threat.
- $P(E_i)$: Total probability of observing the evidence.
- $P(T_i|E_i)$: Posterior probability (threat confidence after evidence is observed).

Table 4: Bayesian Inference

Threat Type	P(Ti)	P(Ei|Ti)	P(Ei)	P(Ti|Ei)
T1 (Phishing + LOLBIN)	0.20	0.91	0.22	0.820
T2 (RDP Port Modification	0.10	0.95	0.12	0.778

To quantifythe probability that an observed event pattern (E_i) implies an actual attack (T_i), the authors calculated the Threat Event probability with Bayesian Inference model, where:

- T_1 = Phishing-based attack with Lolbin
- T_2 = RDP Port manipulation
- E_1, E_{13}: observed Sysmon events
- $P(T_i|E_i)$: posterior probability of attack evidence

Using Bayes' Theorem: $$P(T_i|E_i) = \frac{P(E_i|T_i) \cdot P(T_i)}{P(E_i)}$$ …. Equation 3

with $P(E_1|T_1)$ = 0.91, $P(E_{13}|T_2)$ = 0.95 and Priors $P(T_1)$ = 0.2, $P(T_2)$ = 0.1

Then $P(T_2|E_{13}) = 0.95 * \frac{0.1}{P(E13)}$ ~ Normalized via all events.

The authors also calculated the Persistence Indicator where x = Microsoftupdate.exe found in the Startup folder for f(x)=1

$$f(x) = \begin{cases} 1, & \text{if } x \in \text{Startup Folder} \\ 0, & \text{otherwise} \end{cases}$$ …. Equation 4

- Port Modification validation is $dp = |p_{new} - 3389|$ ~ non-standard RDP port = stealth vector.
- Lateral Movement risk score with weights: $R = \alpha \cdot N_{conn} + \beta \cdot N_{FTP} + \gamma \cdot N_{reg}$ … Equation 5

Table 5 presents the Threat Confidence Score (TCS) for the phishing-based attack using a malicious Microsoft Word document provides a quantified representation of the threat's severity based on observed system behaviors. In this case, the parent process Winword.exe spawned a known living-off-the-land binary (LOLBIN), cmd.exe, which proceeded to download a suspicious payload (MicrosoftUpdate.exe) from a GitHub URL. The activity was traced through a chain of Sysmon event codes: 1 (process creation), 11 (file creation), 3 (network connection), and 5 (process termination). Each of these events contributes to the threat's behavioral complexity.

Table 5: Threat Confidence Score

Threat Hunt	Trigger Process	Malicious Binary	Event Codes& Hits	Key Artifact	Persistence	Exfiltration
Phishing via Word Document	Winword.exe	cmd.exe	1, 3, 5, 11 / 44	MicrosoftUpdate.exe	Startup Folder	Port 9000
RDP Port Modification	Winword.exe	reg.exe	1, 3, 13, 4624 / 20	RDP Port: 3398	Registry Key	FTP Port 21

TCS is calculated as a linear weighted model $\text{TCS} = \omega_e \cdot n_e + \omega_p \cdot \delta_p + \omega_x \cdot \delta_x$ …. Equation 6 were:

- n_e=4 is the number of distinct event codes
- δ_p=1indicates confirmed persistence (via file creation in the Startup folder)
- δx=1 indicates confirmed exfiltration (via suspicious internal communication over port 9000).

Applying the weights ω_e=2.5, ω_p=5, and ω_x=5,the resulting TCS = (2.5×4)+(5x1)+(5x1) =20.

This high score reflects both the malicious intent and tactical execution breadth. TCS enables security operations teams to prioritize incidents not only by detection volume but also by behavioral depth, combining evidence from process lineage, persistence mechanisms, and

outbound connections to assign accurate severity ratings. This score can be used in SIEM correlation rules and incident triage workflows.

3. Entropy Analysis

Entropy analysis quantifies the uncertainty in a dataset by calculating the probability distribution of observed events, like Sysmon event codes from the threat hunt logs. Incorporating entropy analysis into the threat hunt validation process adds a powerful statistical lens to traditional event-based detection. By measuring the unpredictability in system logs, entropy identifies patterns not easily visible through deterministic rule matching. Using Shannon entropy, each event code's frequency was converted into a probability, then summed using:

$H(X)=\sum p(x_i)\log_2 p(x_i)$…. Equation 6

where:

- $H(X)$: Entropy,
- $p(x_i)$: Probability of each event code x_i
- n: Total number of unique event codes.

Table 6 presents the Event Code, Count & Probability values calculated from the dataset.

Table 6: Event Code, Count & Probability

Event Code	Count	Probability
event.code:1	120	0.4
event.code:3	80	0.267
event.code:11	40	0.133
event.code:13	30	0.1
event.code:5	20	0.067
event.code:4624	10	0.033

Figure 20 visualizes the probability distribution of event codes for the entropy analysis.

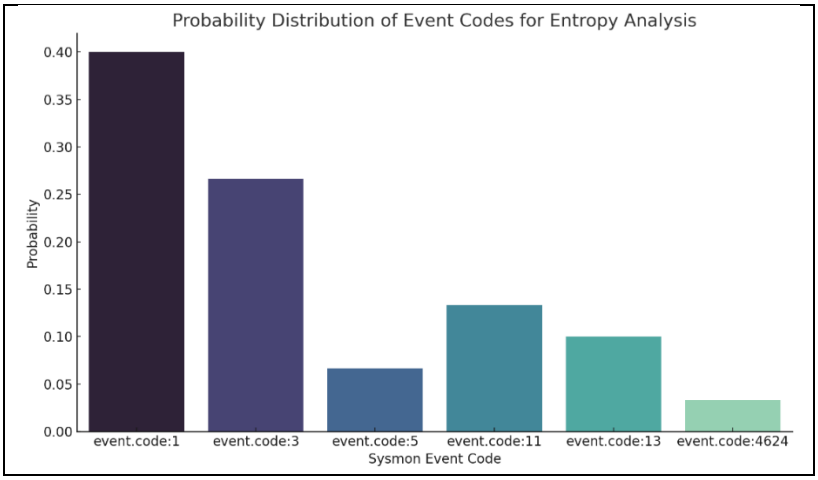

Figure 20: Entropy Analysis

H(X)= 2.181 bits so for the calculated entropy value of 2.181 bits suggests a moderately high diversity of event types, supporting the hypothesis that multiple attack stages occurredranging from execution to persistence and exfiltration. This strengthens the credibility of the threat hunt

findings and helps prioritize further analysis. Ultimately, entropy-driven validation enhances situational awareness, helping SOCs distinguish between benign activity and complex, multi-vector attacks. This provides a scalar value reflecting the variability in system behavior. A higher entropy suggests a more diverse set of events, which often corresponds to complex or anomalous activity. By analyzing entropy, analysts can detect deviations from normal baselines, which may indicate stealthy or distributed attacks in progress:

- Entropy measures unpredictability - In the context of cybersecurity, higher entropy indicates greater variability in the types of system events being observed. This typically suggests complex behavior, possibly driven by a combination of legitimate processes and malicious activity.
- Scale interpretation - Entropy is measured in bits and ranges from:
    - 0 bits: Absolute certainty (e.g., only one type of event).
    - $\log_2(n)$ bits: Maximum entropy for n equally probable events.

In our case, with 6 event codes the maximum entropy = $\log_2(6) \approx 2.585$ bits

For this research dataset the value 2.181 bits (~84% of max) suggests the event activity is not uniform as some events (like event.code:1) occur more often, but it's not skewed either: multiple other event types are meaningfully present or, this diversity is often characteristic of multi-stage attacks (execution → persistence → lateral movement → exfiltration).

6. CONCLUSION

The threat hunting efforts described in this research focused on detecting malicious activities involving Microsoft Winword.exe and manipulation of Remote Desktop Protocol (RDP) port settings. The initial threat hunt found that Winword.exe launched the command-line process cmd.exe, which then downloaded a potentially malicious payload, MicrosoftUpdate.exe, from a GitHub repository. This payload established persistence on the compromised system by creating scheduled tasks and placing executable files in the Windows Startup folder, ensuring the malicious code runs after a system restart. Additionally, unusual network activity was observed, including connections to an internal system over a non-standard port 9000, indicating possible lateral movement or reconnaissance within the network.

The second threat hunt focused on detecting unauthorized changes to RDP port settings via registry key modifications. It was observed that Winword.exe indirectly triggered reg.exe to modify the registry entry that controls the RDP listening port, allowing adversaries to evade detection by moving the service to a less-monitored port. This stealth technique was confirmed by successful remote authentication events and supported by inbound firewall connections. Of particular concern was the presence of outbound network traffic, including data exfiltration via FTP port 21, showing that attackers had established persistent, stealthy access and were actively extracting sensitive information from the environment.

Together, these threat hunts highlight how attackers are evolving their tactics to exploit trusted applications and system features for malicious aims. The findings emphasize the vital need for proactive, intelligence-led threat-hunting practices in cybersecurity. By using advanced detection tools like Elasticsearch-based SIEM platforms and detailed query investigations, organizations can spot subtle signs of compromise that traditional security controls might miss. Ultimately, this research shows that ongoing threat-hunting is crucial for uncovering sophisticated attack methods and reducing risks associated with living-off-the-land binaries, registry manipulations, and data exfiltration, thereby strengthening organizational resilience against persistent and advanced cyber threats.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] "What Is Threat Hunting? | A complete guide | Zscaler," www.zscaler.com. https://www.zscaler.com/zpedia/what-is-threat-hunting.

[2] "What Are Advanced Persistent Threats? | IBM," www.ibm.com. https://www.ibm.com/topics/advanced-persistent-threats.

[3] 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, Jan. 2022, doi: https://doi.org/10.3390/electronics11192992.

[4] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, and L. Mostarda, "Capturing-the-Invisible (CTI): Behavior-based Attacks Recognition in IoT-oriented Industrial Control Systems," IEEE Access, pp. 1–1, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2998983.

[5] D. Hermawan, N. G. Novianto and D. Octavianto, "Development of Open Source-based Threat Hunting Platform," 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2021, pp. 1-6, doi: 10.1109/AiDAS53897.2021.9574308.

[6] Fortinet, "SolarWinds Supply Chain Attack," Fortinet, 2023. https://www.fortinet.com/resources/cyberglossary/solarwinds-cyber-attack.

[7] "REvil Ransomware Attack on Kaseya VSA: What You Need to Know," www.varonis.com. https://www.varonis.com/blog/revil-msp-supply-chain-attack

[8] J. Easterly and T. Fanning, "The Attack on Colonial Pipeline: What We've Learned & What We've Done Over the Past Two Years," Cybersecurity and Infrastructure Security Agency, May 07, 2023. https://www.cisa.gov/news-events/news/attack-colonial-pipeline-what-weve-learned-what-weve-done-over-past-two-years.

[9] A. Goss, "LOLBins (Complete Guide to Living Off the Land Binaries)," StationX, Jun. 26, 2024. https://www.stationx.net/lolbins-living-off-the-land-binaries/ (accessed November 26, 2025).

[10] What is Macro Virus?," www.kaspersky.com, Jan. 13, 2021. https://www.kaspersky.com/resource-center/definitions/macro-virus

[11] "Vietnam CyStack Joint Stock Company," Cystack.net, 2019. https://cystack.net/research/word-based-malware-attack (accessed November 26, 2025).

[12] Imperva, "What is Social Engineering | Attack Techniques & Prevention Methods | Imperva," Learning Center, 2019. https://www.imperva.com/learn/application-security/social-engineering-attack/

[13] FORTINET, "What is Data Exfiltration and How can you prevent it?," Fortinet, 2024. https://www.fortinet.com/resources/cyberglossary/data-exfiltration

[14] Giorgio Bonuccelli, "What is an RDP Attack and How Do You Defend Against Them?," Server and Cloud Blog, Jul. 11, 2022. https://www.parallels.com/blogs/ras/rdp-attack/?srsltid=AfmBOopAcZCKWn3FELOyaqn-9XmQcN-gE_TNR6JOLwSp_vsn4i84TYd2 (accessed November 26, 2025).

[15] B. Gulbay and M. Demirci, "APT-scope: A novel framework to predict advanced persistent threat groups from enriched heterogeneous information network of cyber threat intelligence," Engineering Science and Technology, an International Journal, vol. 57, p. 101791, Sep. 2024, doi: https://doi.org/10.1016/j.jestch.2024.101791.

[16] V. Sachidananda, S. Muneeswaran, L. Yang and K. -Y. Lam, "Do NoT Open (DOT): A Unified Generic and Specialized Models for Detecting Malicious Email Attachments," 2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Exeter, UK, 2023, pp. 412-421, doi: 10.1109/TrustCom60117.2023.00072.

[17] H. Teymourlouei and V. E. Harris, "A Machine Learning Approach to Threat Hunting in Malicious PDF Files," 2023 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, USA, 2023, pp. 782-787, doi:10.1109/CSCI62032.2023.00133.

[18] A. Bhardwaj, Salil Bharany, Ashraf Osman Ibrahim, A. Almogren, Ateeq Ur Rehman, and Habib Hamam, "Unmasking vulnerabilities by a pioneering approach to securing smart IoT cameras through

threat surface analysis and dynamic metrics," Egyptian Informatics Journal, vol. 27, pp. 100513–100513, Sep. 2024, doi: https://doi.org/10.1016/j.eij.2024.100513.

[19] Abbas Kudrati; Chris Peiris; Binil Pillai, "Microsoft Azure Cloud Threat Prevention Framework," in Threat Hunting in the Cloud: Defending AWS, Azure and Other Cloud Platforms Against Cyberattacks , Wiley, 2022, pp.101-182.

[20] N. Soni, M. Kaur, and V. Bhardwaj, "A forensic analysis of AnyDesk Remote Access application by using various forensic tools and techniques," Forensic Science International: Digital Investigation, vol. 48, p. 301695, Mar. 2024, doi: https://doi.org/10.1016/j.fsidi.2024.301695.

[21] S. Toledo, "SSH tunneling to connect to remote computers," Software Impacts, vol. 17, pp. 100545–100545, Jul. 2023, doi: https://doi.org/10.1016/j.simpa.2023.100545.

[22] A. Mahboubi et al., "Evolving techniques in cyber threat hunting: A systematic review," Journal of Network and Computer Applications, pp. 104004–104004, Aug. 2024, doi: https://doi.org/10.1016/j.jnca.2024.104004.

[23] Y. Cohen, D. Hendler, and A. Rubin, "Detection of malicious webmail attachments based on propagation patterns," Knowledge-Based Systems, vol. 141, pp. 67–79, Feb. 2018, doi: https://doi.org/10.1016/j.knosys.2017.11.011.

[24] Sciencedirect.com, 2024.https://www.sciencedirect.com/science/article/abs/pii/S0092656614001123 (accessed Nov 26,2025).

[25] S. Cai, H. Tang, J. Chen, Y. Hu, and W. Guo, "CDDA-MD: An efficient malicious traffic detection method based on concept drift detection and adaptation technique," Computers & Security, pp. 104121–104121, Sep. 2024, doi: https://doi.org/10.1016/j.cose.2024.104121.

[26] A. Bhardwaj, Salil Bharany, A. Almogren, Ateeq Ur Rehman, and Habib Hamam, "Proactive threat hunting to detect persistent behaviour-based advanced adversaries," Egyptian Informatics Journal, vol. 27, pp. 100510–100510, Sep. 2024, doi: https://doi.org/10.1016/j.eij.2024.100510.

[27] E. M. Rudd, R. Harang and J. Saxe, "MEADE: Towards a Malicious Email Attachment Detection Engine," 2018 IEEE International Symposium on Technologies for Homeland Security (HST), Woburn, MA, USA, 2018, pp. 1-7, doi: 10.1109/THS.2018.8574202.

[28] M. Bajer, "Building an IoT Data Hub with Elasticsearch, Logstash and Kibana," 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Prague, Czech Republic, 2017, pp. 63-68, doi: 10.1109/FiCloudW.2017.101.

[29] D. Hermawan, N. G. Novianto and D. Octavianto, "Development of Open Source-based Threat Hunting Platform," 2021 2nd International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2021, pp. 1-6, doi: 10.1109/AiDAS53897.2021.9574308.

[30] H. Almohannadi, I. Awan, J. Al Hamar, A. Cullen, J. P. Disso and L. Armitage, "Cyber Threat Intelligence from Honeypot Data Using Elasticsearch," 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), Krakow, Poland, 2018, pp. 900-906, doi: 10.1109/AINA.2018.00132.

[31] D. Bhatnagar, R. J. SubaLakshmi and C. Vanmathi, "Twitter Sentiment Analysis Using Elasticsearch, LOGSTASH And KIBANA," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.351.

[32] "An Elasticsearch Tutorial: Getting Started," Logz.io, May 28, 2019. https://logz.io/blog/elasticsearch-tutorial/

[33] "Kibana queries and filters | Packetbeat Reference [8.12] | Elastic," www.elastic.co. https://www.elastic.co/guide/en/beats/packetbeat/current/kibana-queries-filters.html

[34] "Lucene query syntax | Kibana Guide [8.12] | Elastic," www.elastic.co. https://www.elastic.co/guide/en/kibana/current/lucene-query.html

[35] IJCNC JOURNAL, "IJNSA 01," International Journal of Computer Networks & Communications (IJCNC), Jun. 16, 2020. https://ijcnc.com/2020/06/16/ijnsa-01-9/.

[36] IJCNC JOURNAL, "IJCNC 06," *International Journal of Computer Networks & Communications (IJCNC)*, Aug. 23, 2025. https://ijcnc.com/2025/08/23/international-journal-of-computer-networks-communications-ijcncvolume-17-number-4-july-2025/.

AUTHORS

Dr. Akashdeep Bhardwaj is working as Professor & Director for the Centre of Cybersecurity at UPES, Dehradun, and is the Chief Executive Officer (CEO) for Global Cybersecurity Association (GCA). An eminent IT Industry expert with over 30 years of experience in areas such as Cybersecurity, Digital Forensics, and IT Operations, Dr. Akashdeep mentors graduate, master's, and doctoral students and leads several projects. Dr. Akashdeep is a Post-Doctoral Fellow from Majmaah University, Saudi Arabia, with a Doctorate (Ph.D.) in Computer Science from UPES Dehradun, a master's in business administration, and an Engineering degree in Computer Science from Pune University. Dr. Akashdeep has published over 200 research works (including Copyrights, Patents, and manuscripts published in highly referred international indexed journals), as well as authored and edited several books & chapters. Dr. Akashdeep worked as a Technology Leader for several multinational organizations during his time in the IT industry.

Dr. Shawon Rahman is a tenured professor of Computer Science and Engineering at the University of Hawaii-Hilo, a Faculty Applied Clean Energy Sciences (FACES) Fellow at the National Renewable Energy Laboratory, and an adjunct faculty member at the University of Missouri–Kansas City. With more than 19 years of teaching experience, he has chaired and supervised numerous Ph.D. dissertations.

Dr. Rahman has secured and managed several federal, state, and foundation grants, including those from NSF, USDA, and DOE. His research spans Cybersecurity, Digital Forensics, Cloud Computing, and STEM outreach. A senior member of IEEE, he has published over 145 peer-reviewed papers and actively serves on editorial boards, professional committees, and national review panels.