






Detecting Unconventional and Malicious Windows Authentication Activities Through Statistical Rarity Assessment

Tarek Radah^{1*}, Habiba Chaoui¹, Chaimae Saadi^{1,2}

¹ Advanced Systems Engineering (ISA), National School of Applied Sciences, IBN TOFAIL University, Kenitra 14000, Morocco

² Laboratory of Systems Analysis, Information Processing and Industrial Management (ASTIMI) EST Salé, Mohammed V University, Rabat 10000, Morocco

Corresponding Author Email: tarekradah@gmail.com

<https://doi.org/10.18280/ijssse.130501>

ABSTRACT

Received: 29 August 2023

Revised: 8 October 2023

Accepted: 17 October 2023

Available online: 10 November 2023

Keywords:

threat-hunting, lateral-movement, intrusion detection, windows authentication, incident response, digital forensics

Threat actors often move laterally through a corporate network to gain access to sensitive data from other machines once they have entered the environment. This is often achieved by using valid and privileged accounts to propagate within the network. However, detecting authentication attempts made by attackers can be challenging for security teams, as these attempts often resemble logons made by users and system administrators. The goal of our research is to develop an approach to identify malicious authentication events on Windows Active Directory environments using statistical analysis. We propose a feature extraction and hashing method applied to events generated by the Windows operating system following a successful logon and conduct statistical analysis to identify rare authentication characteristics that may indicate malicious activity. Our method was applied to a real corporate log with synthetic malicious events and demonstrated the ability to detect malicious authentication attempts effectively. We identified new authentication patterns, some of which were malicious. By using our proposed approach, security defenders can identify and prevent unauthorized access to sensitive data in their network environments.

1. INTRODUCTION

Threat hunting is a process of actively searching for and identifying potential security threats within an organisation's network and system [1]. This is done by analysing data from various sources, such as system logs, network traffic, and endpoint activity. The goal of threat hunting is to identify indicators of compromise (IOCs) and potential security incidents that may have been missed by traditional security measures such as firewalls and antivirus software. Threat hunting on Windows systems involves explicitly analysing data from Windows event logs, registry keys, and other system-level data to identify any unusual or anomalous activity. This can include identifying suspicious file access, network connections, or process execution. Additionally, threat hunting on Windows systems can involve analysing data from endpoint protection software and other security tools that are deployed on the system. One of the significant advantages of threat hunting is the ability to detect advanced persistent threats (APTs) and other stealthy, sophisticated attacks that may evade traditional security measures [2] through what is known as a cyber kill chain, as shown in Figure 1. APTs are a type of cyber-attack that are designed to evade detection and remain undetected on a system for an extended period. By actively searching for these types of threats, organizations can identify and respond to them before they can cause significant damage. Threat hunting can also help organizations meet compliance requirements and reduce incident response time. It allows organizations to identify potential security incidents

early, which can help minimize the impact of an attack and reduce the time and resources required to respond to it.

Additionally, threat hunting enables organizations to identify gaps in their security posture and make improvements to prevent future attacks. Overall, threat hunting is an essential part of an effective security strategy. It should be a regular practice for organizations to detect and respond to potential threats before they can cause significant damage.

Lateral movement is a crucial tactic used by attackers to move deeper into a network and gain access to sensitive information [3]. APT attackers typically use multiple methods to move laterally within a network, such as compromising credentials, exploiting vulnerabilities, using remote access tools, or manipulating malicious insiders. However, all these techniques involve some form of authentication to move through the network [4]. By searching for Windows Event ID 4624 in the security event logs, it is possible to identify successful logins and track the movement of malicious actors within the network.

Event ID 4624 in the Windows security event logs can be used to hunt for lateral movement on Windows systems. Event ID 4624 is a log event generated by the Windows Security Log. It is recorded whenever a user successfully logs on to a computer, and it contains information about the logon session, such as the logon type, the authentication package used, and the logon process [5]. Therefore, event ID 4624 can be used to detect attacks by analysing the information contained in the log event. For example, suppose an attacker is attempting to gain unauthorized access to a system. In that case, they may

use a different logon type or authentication package than what is typically used by legitimate users. Therefore, it is possible to identify anomalies and potential attacks by comparing the logon information recorded in Event ID 4624 to a baseline of regular logon activity.

Event ID 4624 contains several different pieces of information about a logon session, including:

1. **Logon type:** This indicates the method used to log on to the system, such as through a network, through the console, or via a remote desktop connection.
2. **Authentication package:** This indicates the authentication package used to authenticate the user.
3. **Logon process:** This indicates the process that was used to log on to the system, such as Kerberos or NTLM.
4. **Account name:** This is the name of the user account that was used to log on to the system.
5. **Account domain:** This is the domain of the user account that was used to log on to the system.
6. **Logon ID:** This is a unique identifier for the logon session.
7. **Logon GUID:** This is a globally unique identifier for the logon session.
8. **Username:** This is the name of the user who logged on to the system.
9. **Domain name:** This is the domain of the user who logged on to the system.
10. **Logon server:** This is the name of the server that processed the logon request.
11. **Logon time:** This is the time at which the logon occurred.
12. **Logon security identifier (SID):** This is a unique identifier for the user account that was used to log on to the system.
13. **User security identifier (SID):** This is a unique identifier for the user who logged on to the system.
14. **User account control (UAC) value:** This is a value that indicates the status of the user account, such as whether it is enabled or disabled.

15. **User principal name (UPN):** This is the user's email address or login name.
16. **Workstation name:** This is the name of the computer that the user logged on to.
17. **Service name:** This is the name of the service that was used to log on to the system, if applicable.

However, event 4624 does not contain information about whether the corresponding authentication attempt is malicious or not. Moreover, security analysts could not figure it out without precise contextualization.

The main contribution of this research is to enrich windows event 4624 events with valuable features which can be used to figure out the nature of the authentication event. Furthermore, the proposed features can be used to feed a machine learning model for anomaly and outliers' detection [6]. However, in this research, we used a statistical technique to spot rare events. This works assumes that malicious authentication events have unusual characteristics. Existing threat hunting approaches to address this issue involve looking at failed logons and other subjective criteria, such as unusual login times and locations, suspicious logon types, and changes to privileged accounts. However, this approach relies on criteria that are not applicable to every organization. For example, login attempts are made by administrators and IT solutions, and each organization has its own unique IT solutions and administrative practices. As a result, authentication trends will vary across organizations.

This paper is organized as follows: We review the literature on threat hunting in Windows environments in the first section and proposed methods for lateral movement detection. In the second section, we describe our data gathering methodology. Next, we present a detailed explanation of the rarity features proposed by our method and explain how we assign a rarity score to each feature. Finally, we evaluate our approach and provide future perspectives.

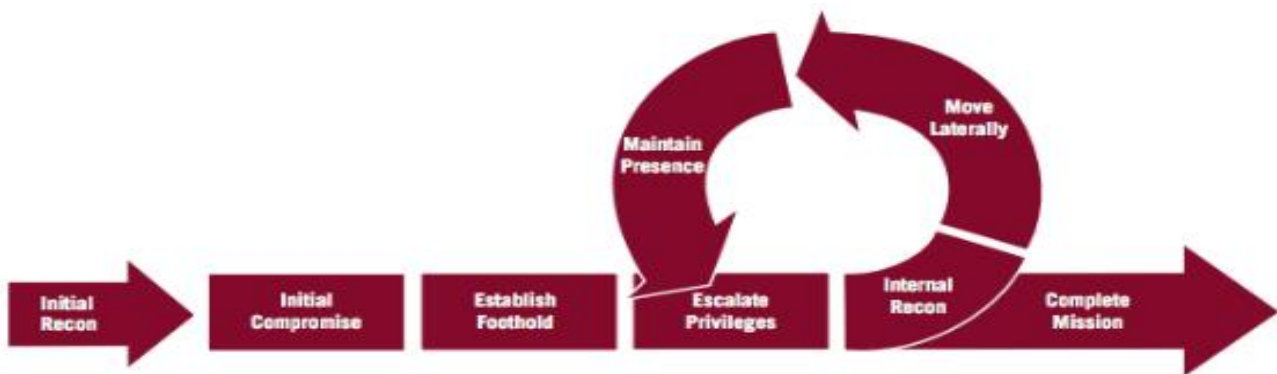


Figure 1. FireEye cyber kill chain

2. RELATED WORKS

This section presents a summary of the relevant literature related to the topic. There is significant scientific research on Windows threat hunting, focusing on developing and evaluating new techniques and methods for detecting and mitigating various types of malwares and other malicious activity on Windows systems. Dwyer et al. [7] proposed a method for detecting anomalies in Windows event logs utilizing the standard deviation. They employed SQL queries to collect data and to compute the average number of events of

a specific type, during any time of the day, for any server or user in the dataset. With this approach, it is possible to establish the average number of events of a specific type and to calculate the standard deviation of those events. They did not rely solely on event ID 4624 (successful logon event) and ignored most of the authentication properties. Consequently, their approach did not consider the different authentication parameters, such as the type of logon and the source of the authentication. They considered all other fields as unnecessary for their research. Our research focuses on spotting rare authentication events based on the authentication properties

aspect. Bowman et al. [8] proposed a method for identifying the lateral movement of Advanced Persistent Threats (APTs) within enterprise level computer networks through the use of unsupervised graph learning. Their technique involves creating an abstract, behaviour-based graph data model that focuses on the specific behaviour of interest. They represented authentication events as graphs and applied unsupervised graph learning to identify low-probability links, thereby detecting anomalous authentications. In their research, instead of relying on Windows-specific authentication events, they used generic authentication logs. Bai et al. [9] presented a technique for identifying RDP-based lateral movement using machine learning. Their approach involves combining multiple datasets and incorporating red team events into them. They then created a synthesized dataset that accurately reflects attack models. To detect malicious log entries, they applied various machine learning algorithms such as Decision Trees (DT), Random Forest (RF), Feed-forward Neural Networks (FNN), Gaussian Naive Bayes (GNB), and Logistic Regression (LB), and identified relevant features. This research focuses on RDP-based lateral movement, that is one of many techniques used by threat actors to spread through a network. Smiliotopoulos et al. [10] explored the ability to detect lateral movement attacks relying on the Sysmon tool in a Windows environment. They proposed a Sysmon configuration covering rules to detect various implementations of lateral movement attacks. Their research was focused on detecting artefacts of execution of lateral movement tools and exploits generated by Sysmon sensors. Their detection was built on artifacts generated by known tools, so their method may fail to detect attacks with new or custom-developed tools by sophisticated threat actors. Ho et al. [11] proposed a system for detecting LM attacks. The system tracks login activity and creates a graph of related logins among hosts to identify anomalies in login patterns, thus allowing for detecting LM attacks. Berady et al. [12] proposed a threat-hunting model that examines Sysmon logs from both the perspective of an attacker and a defender. This approach aims to improve proactive threat detection by using indicators of compromise. However, the model may produce a high rate of false positives if Sysmon is not configured with configuration rules.

Relying on Sysmon for detecting lateral movement attacks can be convenient in some cases, but it is unsuitable for production environments. This is because Sysmon generates a large amount of data that can negatively impact system performance. Additionally, configuring and maintaining Sysmon can be challenging. To effectively use Sysmon, system administrators should tune the configuration file for each server based on the asset's activity and maintain and

update the configuration as needed if the system's function changes.

Additionally, during an incident response event, Sysmon data and other advanced logging capabilities may not be readily available for use by security analysts for digital investigations. This is because Sysmon and advanced logging capabilities are not enabled by default on Windows operating systems and would need to be set up and configured in advance to be available during an incident response.

Conversely, Windows event 4624 is enabled by default on the Windows system and does not require additional configuration or tuning.

While previous work has consisted of applying machine learning algorithms to generic authentication logs, we propose using a feature hashing approach for Windows event 4624. This will allow us to identify rare events and enable security analysts and threat hunters to inspect new authentication patterns inside a corporate network.

3. DATA GATHERING METHODOLOGY

Authentication data was generated by a corporate Windows network. This research was conducted as shown in Figure 2, by gathering corporate Windows authentication events with multiple servers and workstations, generating hundreds of authentication events each. These events are generated every time a user, application, or administrator logs in to perform a specific task. The data used for this research is collected from a production environment and has been anonymized for research purposes.

The environment setup was as follows:

1. Windows logs are collected from the Windows servers and stored in the Windows Event Collector (WEC) [13] server. In the WEC server, events are filtered to include only relevant event logs.
2. The filtered event logs are then forwarded from the WEC server to an OpenSearch [14] cluster using the Winlogbeat [15] agent. OpenSearch is a search engine based on the Lucene library. OpenSearch is forked from Elasticsearch [16]. It is used for handling large amounts of data, such as log analytics, real-time application monitoring, and clickstream analysis.
3. The rarity calculator fetches data from the OpenSearch cluster and performs rarity calculations.
4. The rarity calculator then updates the authentication logs in the OpenSearch cluster with the calculated rarity scores. The method for calculating rarity scores will be explained in a later section of the paper.

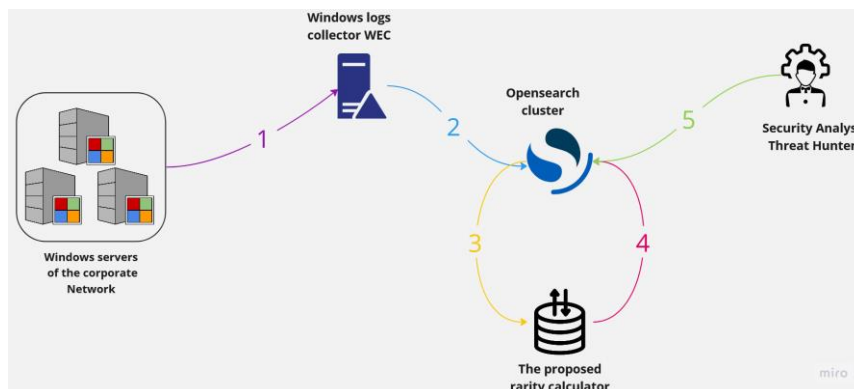


Figure 2. Architecture schema of the lab setup

5. Security analysts can then rely on the rarity scores attributed to authentication logs to hunt for unusual and malicious logons. Security analysts can start by inspecting and investigating authentication attempts with higher rarity scores to determine the purpose of the login event. They can also develop a rarity score threshold based on the average rarity scores in the collected dataset.

4. DEFINING THE DIFFERENT RARITY ASPECTS OF A WINDOWS AUTHENTICATION EVENT

We proposed a different set of rarity aspects to characterize an anomalous Windows authentication event; each one will be represented as a hash of combined Windows events fields related to the characteristics. The hashing algorithm that will

be used for this research is MD5. Despite its vulnerabilities, it was chosen due to its speed and simplicity of calculations, as the algorithm's robustness is not required for this purpose. Each hashing combination will be used to characterize rarity in a different aspect. An overview of the chosen rarity aspect for this research. Table 1 describe the different rarity features.

After generating the RA hashes, each feature (RA hash) will be assigned a rarity coefficient. Finally, the rarity coefficients of all aspects will be evaluated for a given event and based on this, suspicious events will be identified. Since each rarity aspect can indicate a novel authentication pattern, analysts can choose their own threshold based on what they consider to be a legitimate pattern. Although an average rarity threshold can be established, we recommend analyzing the rarity trends first before determining a threshold.

Table 1. Description of the proposed rarity features

Rarity ID	Fields	Rarity Aspect
RA01	<ul style="list-style-type: none"> Logon Type Target Username 	Authentication of a user account with a specific type of connection: High rarity coefficient may indicate that the user does not regularly use this type of logon.
RA02	<ul style="list-style-type: none"> Logon Type Computer Name 	Authentication from a machine with a specific authentication mode: High rarity coefficient may indicate that the corresponding server is not usually reached through that logon type (Exp: RDP, interactive, Network, etc.).
RA03	<ul style="list-style-type: none"> Workstation Name Computer Name 	Authentication from a source machine to a destination machine: High rarity coefficient may indicate that the two machines does not regularly communicates.
RA04	<ul style="list-style-type: none"> Workstation Name Computer Name Logon Type 	Authentication from a source machine to a destination machine with a specific connection type. High rarity coefficient may indicate that the two machines does not regularly communicates using the corresponding logon type (Exp: RDP, interactive, Network, etc.).
RA05	<ul style="list-style-type: none"> Workstation Name Computer Name Target Username 	Authentication from a source machine to a destination machine with a specific user account. High rarity coefficient may indicate that the corresponding user is not regularly used to make connection between the two machines.
RA06	<ul style="list-style-type: none"> Workstation Name Computer Name Target Username Logon Type 	Authentication from a source machine to a destination machine with a specific connection type and user account. High rarity coefficient may indicate that the corresponding user is not regularly used to make connection between the two machines using the specific logon type (Exp: RDP, interactive, Network, etc.).
RA07	<ul style="list-style-type: none"> Logon Process Target Username 	Using an authentication method with a specific user account. High rarity coefficient may indicate that the corresponding user do not regularly use the corresponding type of authentication (Exp: NTLM, Kerberos, Advapi, etc.).
RA08	<ul style="list-style-type: none"> Logon Process Computer Name 	Using an authentication method from a specific source. High rarity coefficient may indicate that the corresponding machine is not regularly reached through the specific type of authentication (Exp: NTLM, Kerberos, Advapi, etc.).
RA09	<ul style="list-style-type: none"> Target Username Computer Name 	Authentication of a user from a specific machine: High rarity coefficient may indicate that the corresponding user do not regularly logon into the corresponding machine.

The code which is responsible for calculating the mentioned hashes is as follows:

```
for log_line in windows_authentication_log_file:
    log_line += {'RA_01': MD5(LOGONTYPE, TARGETUSERNAME)}
    log_line += {'RA_02': MD5(LOGONTYPE, COMPUTERNAME)}
    log_line += {'RA_03': MD5(WORKSTATIONNAME, COMPUTERNAME)}
    log_line += {'RA_04': MD5(WORKSTATIONNAME, COMPUTERNAME, LOGONTYPE)}
    log_line += {'RA_05': MD5(WORKSTATIONNAME, COMPUTERNAME, TARGETUSERNAME)}
    log_line += {'RA_06': MD5(WORKSTATIONNAME,
```

```
COMPUTERNAME, TARGETUSERNAME, LOGONTYPE)}
    log_line += {'RA_07': MD5(LOGONPROCESSNAME, TARGETUSERNAME)}
    log_line += {'RA_08': MD5(LOGONPROCESSNAME, COMPUTERNAME)}
    log_line += {'RA_09': MD5(TARGETUSERNAME, COMPUTERNAME)}
```

5. ATTRIBUTING RARITY SCORE TO EACH RARITY ASPECT OF THE WINDOWS AUTHENTICATION EVENTS

The probability mass function (PMF) is calculated for each aspect of authentication, specifically the RA hash, to

determine if it is a rare event type. The PMF, represented as $p(x) = P(X = x)$, describes the probability of a discrete random variable taking on a particular value and is used to represent the distribution of a discrete random variable. Based on the calculated PMF, a quantile is used to determine if the hash is rare or not. Then, a cumulative distribution function (CDF) is created for the authentication events that belong to the rare class. The CDF, represented as $Fx(x) = P(X \leq x)$, describes the probability of a random variable X being less than or equal to a certain value x. It is a non-decreasing function that ranges from 0 to 1. The CDF score is used to determine the final rarity score for the event hash. This operation is applied to all RA hashes.

The PMF and CDF are statistical approaches that can be used to calculate rarity scores for a variety of events. They are both based on sound mathematical principles, so they can provide very accurate estimates of the rarity of an event. The PMF and CDF can be used to calculate the rarity of events for a wide range of probability distributions, including discrete and continuous distributions. The PMF and CDF are also relatively easy to interpret, so they can be used by people with a variety of backgrounds.

The main function takes a stream of RA hashes and return the associated rarity scores.

Below is the algorithm for rarity calculation:

```

function get_rare_authentication_aspects(array_of_hash_stream)
  for each ra_hash in array_of_hash_stream do
    pmf_of_hash = calculate_pmf_of_ra_hash(ra_hash)
    rare_hash = classify_pmf_using_quantile(pmf_of_hash)
    cdf_of_rare_hash = calculate_cdf_of_rare_ra_hash(ra_hash)
  return (pmf_of_hash, rare_hash, cdf_of_rare_hash)
  end for
end function

```

The resulting rarity score are attributed to each authentication event. Our proposed method involves extracting Windows authentication events with EventID 4624 from a SIEM or log repository. For each of these events, we calculate nine RA hashes, each describing a different aspect of the authentication. We then calculate the rarity score for each RA hash of each Windows authentication event. Authentication events that have high rarity scores across all

authentication aspects are considered rare and should be investigated further by a security analyst. The rarity hash calculation flow is represented in Figure 3. By using the rarity scores, the security analyst can more easily identify potentially malicious Windows authentication events that require closer attention.

Windows authentication enrichment process workflow:

1. Raw windows authentication event:

Logon Type	Target Username	Computer Name	Workstation Name	Logon Process
------------	-----------------	---------------	------------------	---------------

2. Windows authentication event enriched with RA hashes:

Logon Type	Target Username	Computer Name	Workstation Name	Logon Process	RA1	RA2	...	RA9
------------	-----------------	---------------	------------------	---------------	-----	-----	-----	-----

3. Rarity score attributed to RA hashes:

Logon Type	Target Username	Computer Name	Workstation Name	Logon Process	RA1 score	RA2 score	...	RA9 score
------------	-----------------	---------------	------------------	---------------	-----------	-----------	-----	-----------

Regarding limitations, event 4624 is always logged unless it is disabled by default. However, false positives may be raised if a new legitimate login pattern appears, especially when a new IT solution that performs authentication is adopted. Security analysts can verify these false positives. However, once a new pattern is detected as rare, similar events that follow will have lower rarity scores, which may lead to some malicious authentication attempts being considered normal.

Unlike other malicious activities, security analysts often do not know where to start when spotting unusual login attempts during threat hunting. In this case, they can start by investigating authentication attempts with higher rarity scores.

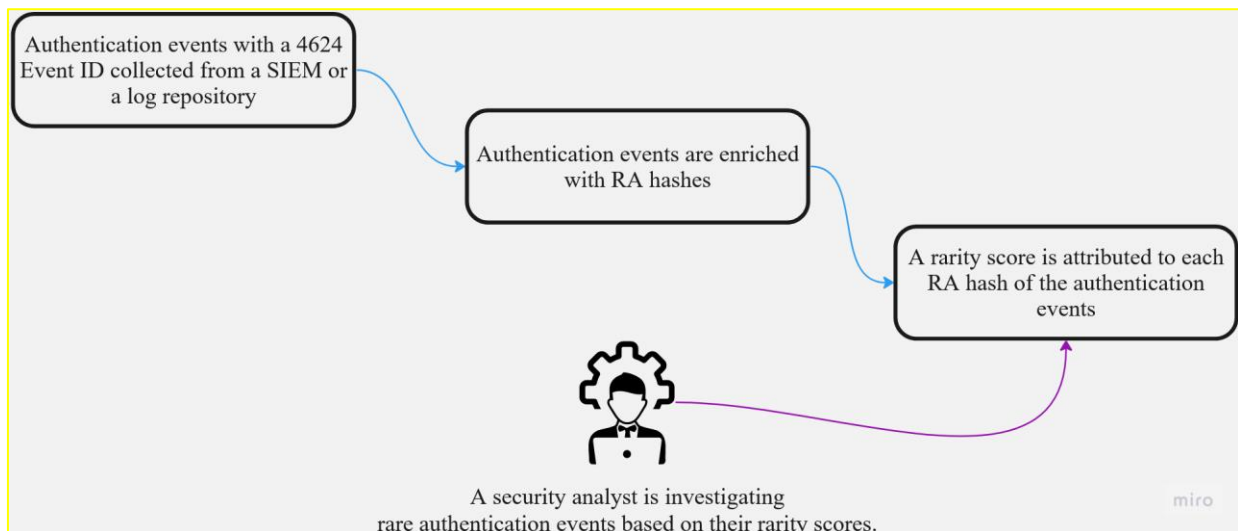


Figure 3. The proposed system workflow

6. EXPERIMENTAL MODEL AND RESULTS EVALUATION

For our experimental model, we combined legitimate authentication data gathered from a corporate network as shown in Figure 4 (as described in section 3 of the article) with synthetic malicious authentication data as shown in Table 2. Our goal is to demonstrate how our proposed approach can effectively identify malicious authentication events.

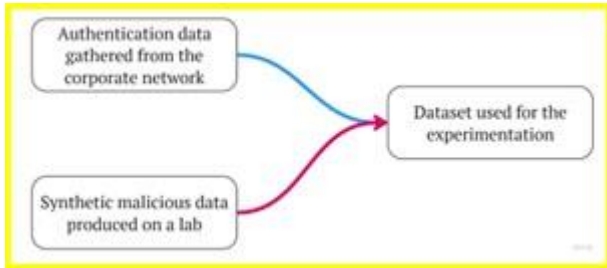


Figure 4. Experimental dataset generation

Our approach assumes that malicious authentication events are relatively rare since threat actors typically do not follow the usual behaviour of users and system administrators. Therefore, by detecting rare authentication events, we can identify potential threats.

Table 2. List of the selected lateral movement technique for the experimentation

Lateral Movement Technique / MITRE Attack Technique ID	Top Threat Actors Utilizing the Technique
Lateral movement using RDP protocol [17] / T1021.001	APT29, APT41, FIN6, Lazarus Group
Remote service creation (PSEXEC) [18] / T1021.002	APT29, FIN6, Turla, Wizard Spider, OilRig, Carbanak, HAFNIUM
Windows instrumentation management (WMI) [19] / T1047	Agent Tesla, APT29, Emotet, Lazarus Group, MuddyWater

To generate synthetic malicious authentication data, we simulated advanced persistent threat (APT) attack techniques in a lab network. These authentication events were involved in lateral movement techniques, which attackers use to propagate inside the compromised network by moving from one machine to another.

The simulation of APT attacks was based on common knowledge of their mode of operation according to the MITRE

Table 5. Evaluation of the proposed method against the injected red team events

Attack Strategy	Count of Injected Simulation Events	Count of Events Detected as Completely Rare	Count of Events Detected as Partially Rare	Count of Undetected Events
Lateral movement using RDP protocol from a compromised public facing application to domain controller.	3	2	0	1
Ransomware distribution from a domain controller to all the accessible Windows server using WMI	31	3	25	3
Spreading across machines using remote service creation	14	6	8	0

Framework. These simulations were conducted in a Windows environment using exactly the same techniques and tools used by advanced threat actors. The names of machines, accounts, and IP addressing schemes are similar to those of the real evaluation environment. The logs generated as a result of these attack simulations were injected into the logs collected from the test environment to generate the final dataset.

We selected the top lateral movement techniques abused by threat actors. Table 3 describes the number of injected events for each attack category.

We generate related authentication events for each lateral movement technique. A paper by Japan CERT [20] details the various artefacts of authentication events generated by each technique. While there are various lateral movement techniques, they all involve an authentication that generates a 4624 event.

When creating authentication events related to lateral movement techniques, usernames, source, and destination machines were randomly selected from the existing legitimate dataset. However, other fields, such as the logon type and logon process, were specific to the simulated technique. Simulation authentication attempts were also distributed across the dataset timespan.

Events related to the following attacks were generated (Table 3).

Table 4 describes the results obtained following the experimentation.

For each attack category, we calculated the completely rare, partially rare and undetected events as shown in Table 5.

Table 3. Simulation events injected into dataset

Attack Strategy	Count of Malicious Simulated Events
Lateral movement using RDP protocol from a compromised public facing application to domain controller.	3
Ransomware distribution from a domain controller to all the accessible Windows servers using WMI.	31
Spreading across machines using remote service creation.	14

Table 4. Classification of results obtained following the experimentation

Total Events	1034856
Completely rare events	53
Partially rare events	20251
Non rare events	1014552

The experimentation was conducted using 1034808 authentication events collected from over 30 Windows machines over a period of two days.

We classified results in three main categories:

- **Completely rare events:** Events with high rarity scores in all aspects (All RA hashes are rare)
- **Partially rare events:** Events with high rarity scores in some aspects (Some of the RA hashes are rare)
- **Non-rare events:** Events without any rare authentication aspect (None of the RA hashes associated with the events are rare)

Out of the 1034856 authentication events, 53 were identified as completely rare, representing only 0.005% of the events. In normal circumstances, rare events would be flagged for further investigation by analysts, as this can save time by eliminating the need to review all events. Of the 53 identified rare events, 11 were found to be related to the red team data injected.

The remaining 42 completely rare events that were not linked to the simulation data were investigated and reviewed by system administrators to determine their validity. They concluded that 19 of the events were uncommon and warranted further investigation, while the remaining 23 events were determined to be false positives, while the remaining 30 events were true positive.

False negative events are malicious events that were not detected as rare by our detector. In our analysis, we considered only events for which all corresponding RA hashes were rare to be rare events. While this increases the number of false negatives, it reduces the number of false positives. Depending on their specific needs, security analysts in any organization can choose their own threshold for what is considered rare. For this experiment, we used the highest rarity threshold.

To evaluate the proposed method, we calculated the FP (False Positive), TP (True Positive), FN (False Negative), the Precision, the Recall and the Accuracy. The Precision is given as follows:

$$Pr = TP / (TP + FP)$$

The Recall is calculated using the following formula:

$$R = TP / (TP + FN)$$

While the Accuracy is given as:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

The results of the evaluation are represented in Table 6 and Table 7.

Table 6. Evaluation of the proposed method

FP	23
TP	30
FN	37
TN	1034766
Precision	0,56
Recall	0,44
Accuracy	0,99

The security analysts time is an extremely valuable resource, and for that we reduced FP by considering only events with

high rarity coefficient in all the rarity aspects. Regarding the FN, we observed that once a rare event occurs, subsequent occurrences of the same event are not always detected as rare by the detector. A high FN rate is not problematic in this case since similar events of the FN class were already identified. This means that security analysts should actively search for similar events in the dataset to identify all instances of rare events. A K-NN ML [21] model could be used to aid this process, but this falls outside the scope of the current research.

Table 7. Confusion matrix

30	23
37	1034766

7. CONCLUSIONS

We propose a human-in-the-loop security detector to identify malicious authentication events, based on the assumption that these types of events are relatively rare. Malicious events and intrusions are by nature rare. After an intrusion, attackers tend to behave differently from internal solutions and system administrators. Events generated by an intrusion are unpredictable and can be difficult to detect using traditional methods. Network intrusion detection can also be considered as a rare event detection problem, as the number of malicious events is typically much smaller than the number of normal events. Our proposed solution involves using nine different RA hashes to enable security analysts to detect rare authentication events from different perspectives. For example, RA03 can be used to detect authentication between two machines that have never communicated before, while RA09 can be used to identify connected users on a particular machine that has not been seen before. Our approach allows security analysts to proactively hunt for threats and take pre-emptive measures to mitigate potential security risks, while reducing the amount of time they need to spend on routine tasks by focusing only on rare and unseen authentication behaviours. The proposed RA hashes can be effectively integrated with other intrusion detection systems, enhancing their overall performance, as the study of Isife et al. [22]. In general, AI and machine learning can significantly enhance threat detection capabilities [23].

REFERENCES

- [1] Wafula, K., Wang, Y. (2019). Carve: A scientific method-based threat hunting hypothesis development model. In 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, pp. 1-6. <https://doi.org/10.1109/EIT.2019.8833792>
- [2] Tatam, M., Shanmugam, B., Azam, S., Kannoorpatti, K. (2021). A review of threat modelling approaches for APT-style attacks. Heliyon, 7(1): e05969. <https://doi.org/10.1016/j.heliyon.2021.e05969>
- [3] Li, M., Huang, W., Wang, Y., Fan, W., Li, J. (2016). The study of APT attack stage model. In 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, pp. 1-5. <https://doi.org/10.1109/ICIS.2016.7550947>
- [4] Bian, H., Bai, T., Salahuddin, M. A., Limam, N., Abou

- Daya, A., Boutaba, R. (2021). Uncovering lateral movement using authentication logs. *IEEE Transactions on Network and Service Management*, 18(1): 1049-1063. <https://doi.org/10.1109/TNSM.2021.3054356>
- [5] Vinaypamrani-Msft. (n.d.). 4624(s) an account was successfully logged on. (Windows 10). 4624(S) An account was successfully logged on. (Windows 10) | Microsoft Learn. <https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/event-4624>, accessed on January 24, 2023.
- [6] Ahmed, M., Mahmood, A.N., Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60: 19-31. <https://doi.org/10.1016/j.jnca.2015.11.016>
- [7] Dwyer, J., Truta, T.M. (2013). Finding anomalies in windows event logs using standard deviation. In 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Austin, TX, USA, pp. 563-570. <https://doi.org/10.4108/icst.collaboratecom.2013.254136>
- [8] Bowman, B., Laprade, C., Ji, Y., Huang, H.H. (2020). Detecting lateral movement in enterprise computer networks with unsupervised graph {AI}. In 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020), pp. 257-268.
- [9] Bai, T., Bian, H., Abou Daya, A., Salahuddin, M.A., Limam, N., Boutaba, R. (2019). A machine learning approach for RDP-based lateral movement detection. In 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, pp. 242-245. <https://doi.org/10.1109/LCN44214.2019.8990853>
- [10] Smiliotopoulos, C., Barmpatsalou, K., Kambourakis, G. (2022). Revisiting the detection of lateral movement through Sysmon. *Applied Sciences*, 12(15): 7746. <https://doi.org/10.3390/app12157746>
- [11] Ho, G., Dhiman, M., Akhawe, D., Paxson, V., Savage, S., Voelker, G.M., Wagner, D. (2021). Hopper: Modeling and detecting lateral movement. In 30th USENIX Security Symposium (USENIX Security 21), pp. 3093-3110.
- [12] Berady, A., Jaume, M., Tong, V.V.T., Guette, G. (2021). From TTP to IoC: Advanced persistent graphs for threat hunting. *IEEE Transactions on Network and Service Management*, 18(2): 1321-1333. <https://doi.org/10.1109/TNSM.2021.3056999>
- [13] Karl-Bridge-Microsoft. (n.d.). Windows event collector - win32 apps. Win32 apps | Microsoft Learn. <https://learn.microsoft.com/en-us/windows/win32/wec/windows-event-collector>, accessed on January 24, 2023.
- [14] About OpenSearch. OpenSearch. (n.d.). <https://opensearch.org/about.html>, accessed on January 25, 2023.
- [15] Winlogbeat Overview. Elastic. (n.d.). https://www.elastic.co/guide/en/beats/winlogbeat/current/_winlogbeat_overview.html, accessed on January 25, 2023.
- [16] What is elasticsearch. Elastic. (n.d.). <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html>, accessed on January 25, 2023.
- [17] Bai, T., Bian, H., Salahuddin, M.A., Abou Daya, A., Limam, N., Boutaba, R. (2021). RDP-based lateral movement detection using machine learning. *Computer Communications*, 165: 9-19. <https://doi.org/10.1016/j.comcom.2020.10.013>
- [18] Ullah, I. (2016). Detecting lateral movement attacks through SMB using bro. Master's thesis, University of Twente.
- [19] Niakanlahiji, A., Wei, J., Alam, M.R., Wang, Q., Chu, B.T. (2020). {ShadowMove}: A stealthy lateral movement strategy. In 29th USENIX Security Symposium (USENIX Security 20), pp. 559-576.
- [20] Center, J.C. (2017). Detecting lateral movement through tracking event logs. JPCERT Coordination Center.
- [21] Zhu, X., Ying, C., Wang, J., Li, J., Lai, X., Wang, G. (2021). Ensemble of ML-KNN for classification algorithm recommendation. *Knowledge-Based Systems*, 221: 106933. <https://doi.org/10.1016/j.knsys.2021.106933>
- [22] Isife, O.F., Okokpujie, K., Okokpujie, I.P., Subair, R.E., Vincent, A.A., Awomoyi, M.E. (2023). Development of a malicious network traffic intrusion detection system using deep learning. *International Journal of Safety and Security Engineering*, 13(4): 587-595. <https://doi.org/10.18280/ijss.130401>
- [23] Fakiha, B. (2023). Enhancing cyber forensics with AI and machine learning: A study on automated threat analysis and classification. *International Journal of Safety and Security Engineering*, 13(4): 701-707. <https://doi.org/10.18280/ijss.130412>