

Detection of Unauthorised System Access Using Native Windows Forensic Artefacts:: A Host-Based Investigation Framework



Student name:	Rajesh Barua	Student ID number:	2513042
Programme:	MSc Computing portfolio		
Module:	Research Methods and Professional Development		
Module code:	ACFA7004	Contribution to Overall Module Assessment (%):	70%
Lecturer:	Dr Seena Joseph	Internal Verifier:	Dr Nitheesh Kaliyamurthy
Assignment Title:	Research Proposal	Word count (or equivalent):	4000 words
Submission deadline:	8 th May 2026	Return date of provisional marks & written feedback:	20 working days after presentation

TABLE OF CONTENTS

1	Introduction	3
1.1	Context and Background.....	3
1.2	Research Problem.....	4
1.3	Research Aim	4
1.4	Research Objectives	5
1.5	Significance of the Research	6
2	Literature Review	6
2.1	Introduction to the Literature Review	6
2.2	Windows Event Log Forensics and Structured Investigation	6
2.3	Machine Learning-Augmented Log Analysis	7
2.4	Host-Based Intrusion Detection and Behavioural Analytics.....	8
2.5	Attack Simulation, Dataset Engineering, and Forensic Reproducibility	9
2.6	Summary and Identified Gaps.....	9
3	Research Design and Methodology	10
3.1	Research Philosophy and Design	10
3.2	Research Strategy	12
3.3	Research Method.....	12
3.4	Forensic Investigation Procedure Design.....	12
3.5	Data Analysis and Evaluation	13
4	Research Materials	14
4.1	Research Data	14
4.2	Ethical Considerations.....	14
4.3	Software, Tools, and Hardware	14
5	Expected Outcomes and Limitations	15
6	Project Timeline.....	16
7	References.....	18

1 INTRODUCTION

1.1 Context and Background

The widespread use of Windows-based computing infrastructure by enterprise and public sector organisations has made the operating system a prime target of credential-based attacks, unauthorised authentication attempts and insider threats. Modern studies have recognized that the conventional rule-based defence strategies are no longer sufficient to deal with the amount and complexity of new access violations, and that more adaptive, data-driven detection methods are now necessary [1]. With the growth in the scale and complexity of digital operations, the safety of the authentication processes has become an urgent issue, and organisations are experiencing continuing difficulties in detecting and addressing repeated failed logins, credential abuse, and privilege creep [2]. It is now known that enterprise hosts are going to generate millions of audit events and endpoint detection and response events each-day, and this amount is producing an extremely large body of forensic evidence which must be systematically analysed to isolate malicious behaviour [3].

In this context, Windows operating system has offered inherent capabilities to capture security-relevant events with its Security Event Logging infrastructure. These logs record authentication attempts, account management actions and system changes and have been identified as being critical sources of forensic evidence in digital investigations [4]. The formalisation of structured forensic analysis of Windows event logs was formally codified with the Windows Event Forensic Process (WinEFP) which formalised a systematic procedural approach to identifying and analysing relevant log events [5]. The capabilities of Windows Event Viewer as being based on the XML-based storage capabilities have since been explored to enhance querying and evidence retrieval [6]. Along with event logs, the Windows Registry forms an important source of user activity traces, system configuration data and persistence artefacts that can be used to provide corroborative forensic evidence that can be used to extend the utility of log-based analysis [7].

The last few years have seen a coming together of digital forensics and machine learning, and is a major development in the discipline. Deep learning frameworks such as Bi-LSTM networks and autoencoders have been shown to have the capacity to detect irregular access patterns by analyzing logs of the operating system [8]. Hybrid detection systems comprising memory forensics and Windows Event Log analysis have been shown to have classification accuracies of up to 99% when using Random Forest classifiers [9]. Supervised and

unsupervised machine learning-based dynamic analysis of Sysmon event logs have yielded F1-scores of up to 0.9873 [10], whilst machine learning-based detection of ransomware in Windows Active Directory environments has further confirmed the richness of evidence of native Windows artefacts [11]. These developments are indicative of an evolving recognition that Windows event logs can provide a significant amount of forensic intelligence - but little has been done in practice to exploit this information to detect targeted, host-based unauthorised access.

1.2 Research Problem

The systematic correlation of Windows Security Event Logs with Registry artefacts with the purpose of identifying repeated failed authentication attempts and re-creating timelines of access has received inadequate scholarly interest. As shown by Sajan et al. [4], unsuccessful logon, and access violations are the key indicators of a potential insider threat, although there is no standardised, validated framework of integrating these types of events with Registry-based evidence in a cost-effective, host-based investigation environment. The overall issue of event log correlation between multiple artefact types to rebuild a sequence of multi-step attack attempts with only native Windows tools.

This is a compounding issue because most of the currently available forensic frameworks are designed to operate in an enterprise-scale or law-enforcement focused environment, and they rely on commercial SIEM platforms and forensic tools that are out of reach to resource-limited organisations and in academic settings [2]. Although investigative studies of Registry forensic have shown the investigative utility of Windows hive data to recover evidence of user activity [7], these studies have been targeted at hive extraction and data representation as opposed to specific detection of authentication anomalies. As a result, there exists no academically validated, formalized procedure that combines the analysis of Windows Security Event Logs and the examination of Registry artefacts to detect and reconstruct unauthorised access to Windows systems through login into the Windows Security Event Logs, in the absence of commercial forensic software.

1.3 Research Aim

The aim of this research is to investigate and evaluate the effectiveness of native Windows forensic artefacts - specifically Security Event Logs and Registry entries - in detecting unauthorised system access through a structured, host-based forensic investigation framework

applicable to Windows 10 and Windows 11 environments, without reliance on commercial forensic tools.

1.4 Research Objectives

The following four SMART research objectives operationalise this aim, as summarised in Table 1.

Table 1: Research Objectives and Measurable Outcomes

No.	Objective	Measurable Outcome
1	To critically analyse existing literature on Windows Event Log forensics, Registry artefact investigation, host-based intrusion detection, and event log correlation methods to identify established techniques, theoretical frameworks, and gaps in current knowledge.	Thematic literature review identifying key themes and gaps addressed by the proposed study
2	To design a structured forensic investigation procedure integrating Windows Security Event Log analysis (Event IDs 4624, 4625, 4648, 4740) with Registry hive examination of NTUSER.DAT, SAM, and SYSTEM entries, to correlate evidence of unauthorised authentication activity.	Documented procedure specification covering both artefact types
3	To implement and test the designed procedure through a controlled simulation of repeated unauthorised login scenarios on a Windows 10/11 virtual machine, collecting and processing native artefact data against a known ground truth.	Simulation dataset with complete ground truth and collected artefacts
4	To evaluate the procedure's effectiveness against measurable forensic criteria - detection rate, false positive rate, and timeline reconstruction completeness - and assess its applicability in low-cost, host-based settings.	Quantitative evaluation report with computed metrics

1.5 Significance of the Research

The value of this study is that it added value to both the academic and forensic disciplines. Academically, the gap that was addressed by the study is the lack of a validated and integrated framework to correlate windows event logs and Registry artefacts to identify frequent failed authentication events. This adds novel methodological understanding to the digital forensics discipline, especially to Windows 10/11 settings that are yet to be well-investigated in comparison to their ubiquity in enterprises. In a pragmatic sense, the study encourages affordable low-cost investigation systems to organisations that cannot support enterprise SIEM infrastructure [2]. Detection systems in the literature often need commercial platforms or large-scale labelled datasets that are not necessary to small organisations or academic environments [3]. By showing that native Windows utilities and open-source tools are adequate to the structured forensic investigation, this research provides a directly applicable contribution to the incident responders, cybersecurity educators, and host-based forensic practitioners, including security auditing, incident triage and investigator training across Windows environments where failed logon events are key threat indicators [4].

2 LITERATURE REVIEW

2.1 Introduction to the Literature Review

This literature review discusses the body of scholarship applicable in this context to detect unauthorised access to a system using native windows forensic artefacts. The literature reviewed is published between 2014 and 2026, and is thematically organised around four areas, namely Windows Event Log forensics and structured investigation frameworks; machine learning-augmented log analysis; host-based intrusion detection and behavioural analytics; and attack simulation, dataset engineering and forensic reproducibility. The review not only describes the papers individually but also synthesises the findings, methodologies, and limitations stated across the studies, and concludes by placing the proposed research in the perspective of the identified gaps.

2.2 Windows Event Log Forensics and Structured Investigation

The digital forensics literature has a longstanding history of references to the forensic analysis of Windows Event Logs. Do et al. [5] suggested the Windows Event Forensic Process

(WinEFP), a structured approach that directs practitioners in identifying and analyzing the pertinent log events. Although this framework offers procedural rigour, it is designed to investigate cases of repeated authentication failures after they have occurred and does not allow detection of repeated occurrences of authentication failures, before they happen. This was furthered by Sajan et al. [4], who showed that the logon attempts that failed and access violations recorded in the Windows Security Event Logs are important indicators of a possible insider threat; the study did not take the form of an experiment nor did it involve the use of multi-artefacts. An XML-based event log storage model that was proposed by Patel et al. [6] aimed at a better querying and forensic analysis of the event logs, although the contribution similarly focused on the administration of the event logs over predictive correlation or Registry integration.

The most recent synthesis of this field, the survey of event log correlation methods to multi-step attack detection offered by Shaukat et al. [3] was the most comprehensive since the 2012 survey. However, more importantly, Shaukat et al. admitted that the issues of real-time deployment are not solved yet and that no specific framework can deal with host-based failed login detection without enterprise infrastructure. Khan et al. [12] bridged the gap in reproducibility by developing a synthetic event-log generator and benchmarking framework, which, using the DBSCAN clustering, with a score of above 0.95, showed that controlled experimental settings could be used to generate reliable forensic datasets. These studies all confirm the forensic usefulness of Windows event logs and reveal the central gap: no established procedure can be used to combine event log analysis with Registry artefact analysis to perform the targeted detection of repeated authentication failures in a low-cost, host-based environment.

2.3 Machine Learning-Augmented Log Analysis

A significant amount of research has utilized machine learning on the security logs of a Windows environment, and has steadily enhanced its detection performance at the expense of increasing its resource needs. As demonstrated by Patil and Prabhu [9], a hybrid model that combines memory forensics with windows event logs analysis using random forest classifier yields 99 percent detection accuracy of malware and 95 percent of classification accuracy of event logs based on windows event logs. Achmad et al. [10] determined that the Windows event log data has rich behavioural signatures that can be used to automatically classify the data. Keyogeg et al. [11] generalized this to Active Directory environments, where a Random

Forest model was found to be successful in identifying ransomware privilege escalation and lateral movement patterns in log data, but scaled poorly to actual enterprise conditions. Studiawan [8] made a more general methodology contribution by an automated anomaly detection framework to run system logs using Bi-LSTM networks and autoencoders.

The border of this theme is denoted by explainable AI and large language model methods. The accuracy rates of TabNet and SHAP explainability are 99.97% binary malware detection accuracy [13], although the research was based on malware family classification, as opposed to authentication anomaly detection. Ghosh et al. [14] designed an explainable IDS to classify DoS using EfficientNet-B0 with a high accuracy of 99.99% and an overhead of running a trained model is also a problem. Vaarandi and Bahşi [15] suggested an unsupervised template mining framework based on an LLM on security event log data, demonstrating the ability to extract patterns without any labelled data. Importantly, none of these high-accuracy systems can be used without large labelled training datasets, lots of computational resources, or enterprise-scale infrastructure.

2.4 Host-Based Intrusion Detection and Behavioural Analytics

Research on host-based intrusion detection offers valuable background information to the proposed research. You et al. [16] tested deep learning pipelines to HIDS on system call sequences pre-processed with TextRank, with CNN + BiGRU models, finding that they reached F1-scores of 0.90 to 0.94, but tested them only on specific scenarios and not on generalisability. Mohamed and Arabo [2] proposed a prototype of SIEM-integrated insider threat based on calibrated one dimensional CNN models in the Splunk system with an ROC-AUC of logon events of 0.936, confirming that analysis of logon events can achieve high detection performance - though the system needed enterprise SIEM infrastructure. Amro [17] created a hybrid ransomware detection framework with a stacking ensemble with macro F1 of around 0.93, but the framework relied on a synthetic dataset of 224 features. An AI-based anomaly detection model in embedded systems using PCA and ensemble learning was proposed by Dragushin et al. [18] but the work focused on network traffic as opposed to artefacts of host-based authentication. In all of these studies, a common limitation is the lack of low-cost, validated, and applicable host-based investigation techniques in reachable academic institutions.

2.5 Attack Simulation, Dataset Engineering, and Forensic Reproducibility

The quality and reproducibility of experimental datasets are important to the practical validity of forensic detection research. Landauer et al. [19] proposed a labelled cybersecurity log dataset, CAM-LDS, which is a collection of seven attack scenarios mapped to 81 techniques and 13 ATT&CK tactics, which demonstrates the viability of the controlled simulation of attacks to generate labelled forensic datasets with complete ground truth - a methodology that directly applies to the proposed experimental design. This principle was strengthened by Khan et al. [12], who maintained that in order to perform rigorous benchmarking the synthetic, parametrised, generation of data sets is necessary. Analyses of event logs over a 7-month period [20] have shown that event logs do contain enough signal to be able to perform probabilistic inference, with a belief propagation model achieving a true positive rate of 95.2 with a false positive rate of 0.68 establishing the information-theoretic richness of log data even with minimal feature engineering.

2.6 Summary and Identified Gaps

The literature reviewed confirms that Windows Security Event Logs are well-known forensic tools, which have great evidential value. Machine learning augmented methods have high detection rates but always require labelled data, enterprise infrastructure, and large amounts of compute resources. The host-based HIDS systems are shown to be highly effective when using the system call traces but are not used when dealing with the Windows security events in the accessible environments. Correlation of event logs has been synthesised as central to recent SOC operations, but no validated procedure has so far been published to support the targeted integration of Security Event Logs with Registry artefacts in detecting repeated failed authentication attempts. Although it was established as a discipline of recovering traces of user activity [7], Registry forensics has not been linked to the brute-force detection or access time-line reconstruction in low-cost host-based environments. This paper fills these convergent gaps by designing and experimental validating an integrated forensic investigation process using exclusively native windows 10/11 artefacts and open-source tools.

3 RESEARCH DESIGN AND METHODOLOGY

3.1 Research Philosophy and Design

The study is placed in a positivist philosophical approach which reflects an assumption that Windows Security Event Logs and Registry entries are objective, observable and measurable evidence of system activity, which exists without consideration of research interpretation. Positivism is suitable in the study where the data is collected in a structured way and its performance is evaluated in a quantitative way. The study follows a deductive method, which starts with the theoretical postulation, based on the reviewed literature, that the native Windows forensic artefacts are adequate to identify and recreate the patterning of the unauthorised access to the system without commercial tools. The operationalisation of this proposition is accomplished by the testability of the objectives, as well as by evaluating them against measurable results.

The overall research paradigm is Design Science Research (DSR), which is commonly used in technical fields such as computer science and information systems. DSR can be described as the design and development of artefacts - in this case a structured forensic investigation procedure - as a process of creating knowledge, and resolving the identified problems. The DSR cycle follows problem identification, objective definition, artefact design, development, testing, evaluation and communication of results as shown in Figure 1. This paradigm is suitable since the study generates a methodological artefact the effectiveness of which is empirically tested in a controlled experimental study.

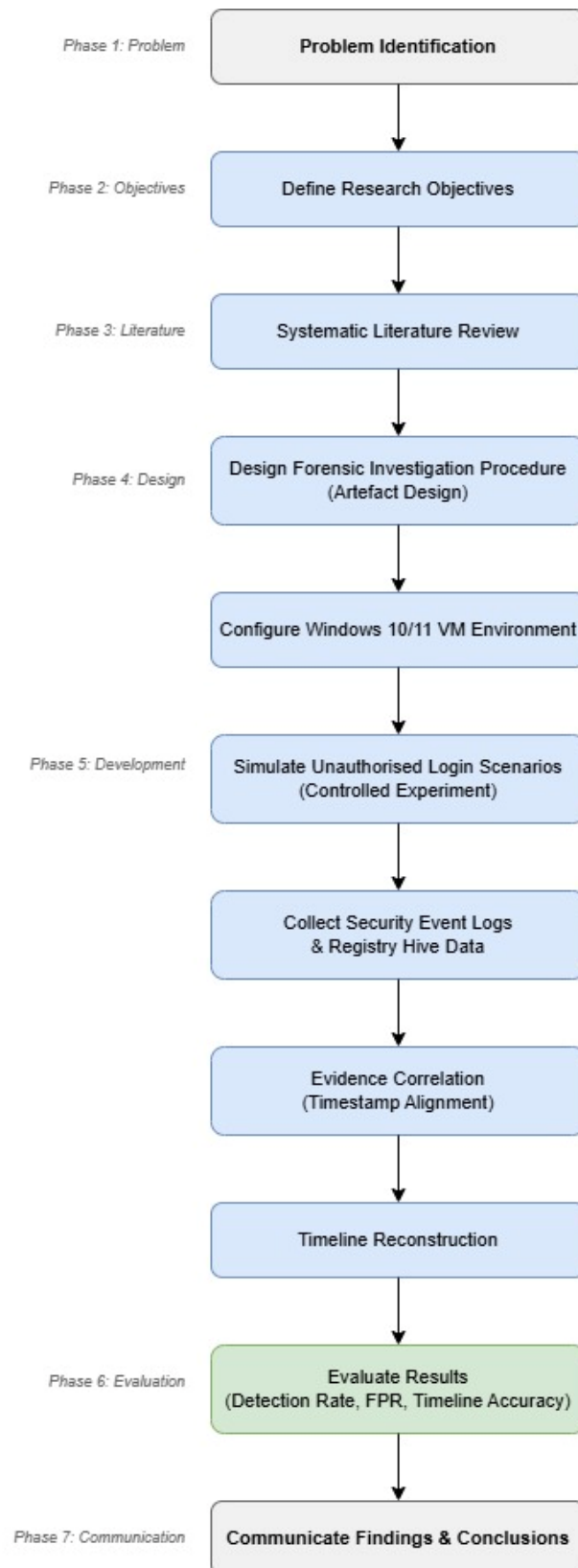


Figure 1: Research Methodology Flowchart (DSR Framework)

3.2 Research Strategy

The study is based on the strategy of controlled experiment. The experimental environment is a virtual machine with Windows 10/11 installed, and the scenarios of repeated unauthorised logins are simulated in the virtual machine. The independent variable is the type and pattern of attempts to log in - including repeated failed logins, account lockouts, and explicit use of the credential - and the dependent variables are the artefacts of the attempt at log in generated by the attempt, measured in terms of presence, completeness and interpretability within the output of the investigation. The controlled VM setup guarantees that the ground truth is completely known, and allows to rigorously assess the detection accuracy and completeness of the timeline reconstruction. This approach is in line with the controlled attack simulation methodology that was employed in similar forensic investigations [19] and directly answers the concerns of reproducibility highlighted by Khan et al. [12].

3.3 Research Method

The research methodology used is a quantitative research, which is in line with the philosophy and methodology of positivism and deductive research. The results of the simulation are analysed with respect to structured parsing and timestamp correlation methods and performance is measured with respect to standard quantitative measures of detection rate, false positive rate and completeness of timeline reconstruction which are consistent with evaluation conventions established in the reviewed literature [9], [10]. This approach is suitable since the question of the study is a measurable amount of forensic capability of a defined artefact to detect a given type of security incident and because the controlled environment means that quantitative results can be compared against known ground truth. The study does not require a mixed method since the research is about the empirical measurement of procedure effectiveness and not a subjective experience of the investigator.

3.4 Forensic Investigation Procedure Design

The investigation process is modelled in five steps as shown in Figure 2. Stage 1 Windows 10/11 VM is set up with audit policies set to create Security Event Logs. In Stage 2, the cases of unauthorised logins are simulated, including: repeated unsuccessful attempts to logon (Event ID 4625), successful attempts of logon (Event ID 4625), explicit credential use (Event ID 4648) and account lockout events (Event ID 4740). Stage 3 uses PowerShell to extract Security Event Logs using the Get-WinEvent cmdlet and Windows Event Viewer, and extracts Registry hive

data using the Get-WinEvent cmdlet and windows event viewer. Using RegRipper and Registry Explorer, DAT, the SAM hive, and the SYSTEM hive. In Stage 4, correlation of evidence is made by mapping timestamps between the two artefact streams in order to discover patterns of cross-artefact evidence that support the possibility of unauthorised access. The correlated evidence is then used in Stage 5 to reconstruct a chronological access timeline, which is compared with the known simulation ground truth.

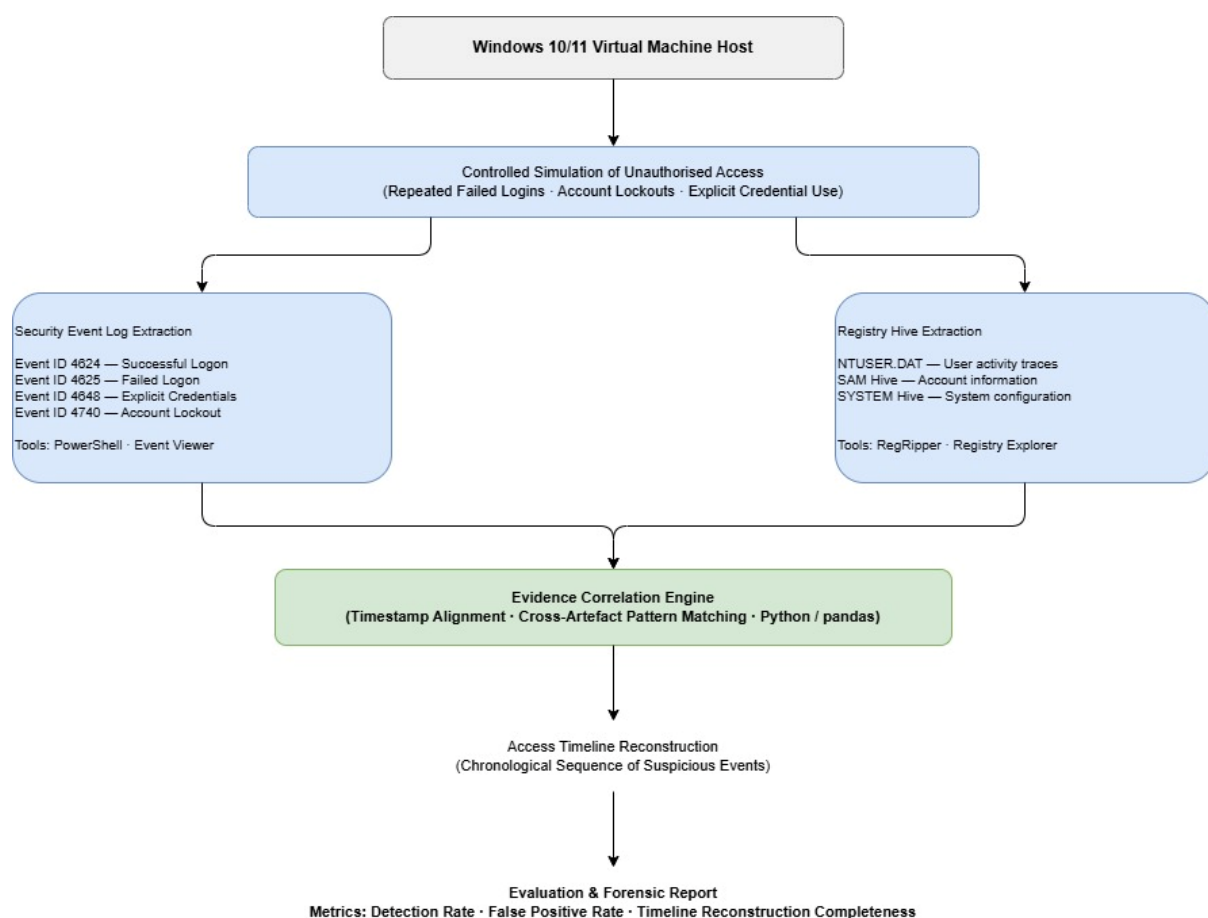


Figure 2: Forensic Investigation Architecture and Data Flow

3.5 Data Analysis and Evaluation

Python with pandas is used to process and format log and Registry data, allowing systematic time alignment and pattern matching of log and Registry artefacts, and timeline visualisation is generated using matplotlib. Three main evaluation measures are calculated: Detection rate, which indicates the proportion of the known access sequence correctly recreated by the correlated evidence; false positive rate, which is the proportion of legitimate events incorrectly flagged by the correlated evidence; and completeness of timeline reconstruction, which is the

proportion of the known access sequence correctly recreated by the correlated evidence. These metrics are consistent with the evaluation standards that are used in similar forensic detection studies [9], [10].

4 RESEARCH MATERIALS

4.1 Research Data

The data to be used in this study is created using controlled simulation in a Windows 10/11 virtual machine-based environment. No prior or external datasets are applied, so there is perfect knowledge of the ground truth and privacy concerns are eliminated because of the use of real organisational data. The simulation creates Windows Security Event log files in the evtx format and Registry hive files in the hive format. The format of DAT that are combined together form the main research data set. All data collection is done on a local machine without any network transfer, thus data security and integrity during data collection are ensured.

4.2 Ethical Considerations

No human subjects, personal information or actual organisational structures are involved with this study. The data are all synthetically produced by a controlled simulation environment. No sensitive or personally identifiable or confidential information is collected, processed or stored at any point of the research. The research has a low level of ethical risk and does not involve any specific institutional ethics committee approval beyond the usual requirements of conducting a research. The simulation environment is segregated in a virtual machine so that there is no interaction between the simulation environment and external systems. A form of ethics has been filled and provided independently through Moodle submission link, as a module requirement.

4.3 Software, Tools, and Hardware

Table 2 summarises all software, analytical software and hardware used in this study. All the tools are freely accessible and do not need any commercial licensing, which helps to directly support the low-cost, accessible design of the investigation framework.

Table 2: Research Software, Tools, and Hardware Requirements

Category	Tool / Resource	Purpose
Virtualisation	VMware Workstation / VirtualBox	Windows 10/11 VM host environment
Log Extraction	Windows Event Viewer; PowerShell (Get-WinEvent)	Security Event Log collection (.evtx format)
Registry Analysis	RegRipper; Registry Explorer	Hive parsing and artefact extraction (.DAT format)
Data Analysis	Python 3.x - pandas; matplotlib	Timestamp correlation and timeline visualisation
Data Export	openpyxl	Structured results export to spreadsheet
Hardware	Personal computer - 8 GB RAM; 50 GB storage	Local data collection; no GPU or cloud resources required

5 EXPECTED OUTCOMES AND LIMITATIONS

It is expected that the proposed forensic investigation procedure will demonstrate a quantifiable ability to identify repeated failed attempts to log-in and rebuild the access timelines using the Windows 10/11 native artefacts alone. Security Event Logs are expected to provide the main signal of detection based on clustering patterns of Event ID 4625 entries, whereas Registry artefacts will be used to provide corroborating evidence of user presence and session activity. It is hoped that the study would result in a documented validated procedure that can be directly applied in the low-cost, host-based forensic investigation environment, which would provide an accessible methodological alternative to the commercially-dependent tool-based approaches.

Theoretical contribution In theoretical terms, the study is likely to prove that the forensic evidential value of native Windows artefacts as theorised in the reviewed literature [4], [6], could be operationalised through the structured, reproducible process without machine learning

or enterprise infrastructure. This result would be a generalisation of the original forensic process models of Do et al. [5] and the multi-artefact correlation principles synthesised by Shaukat et al. [3].

The research has a number of constraints that should be considered. First, it is limited to scope; host-based unauthorised login attempts are within the scope of investigation; network-based intrusions, subsequent lateral movement, and advanced privilege escalation are out of the investigative scope. Second, the process requires that Security Event Logs be enabled and intact; when the audit policies are disabled, or the logs are cleared then forensic evidence will not be available - a known limitation in the forensic literature [5]. Third, Registry artefacts offer corroborating and not primary authentication evidence that cannot be used as a diagnostic resource by itself. Fourth, although ground true, the controlled simulation environment may not be able to generalisably apply to real-world enterprise environments, and future research using operational datasets will be needed.

6 PROJECT TIMELINE

The studies will take the form of a twelve-week architecture based on the four research objectives and the phases of the Design Science Research methodology. The activities in Week 1 and 2 are given priority in terms of the activities to be undertaken in order to support the goal of designing a model to be used to study the impact of the problem under investigation. Weeks 5 and 6 will be dedicated to simulation and data collection activities, with a substantial amount of time devoted to the analysis and writing of the project in the last four weeks. Table 3 shows the 12-week plan.

Table 3: 12-Week Project Timeline

Week	Major Tasks	Milestone
1-2	Finalise and consolidate literature review; complete critical analysis of all twenty confirmed sources	Literature review section complete
3	Design forensic investigation procedure; define Event ID targets (4624, 4625, 4648, 4740) and	Procedure design document complete

Week	Major Tasks	Milestone
	Registry hive scope (NTUSER.DAT, SAM, SYSTEM)	
4	Configure Windows 10/11 virtual machine; enable audit policies; verify artefact generation baseline	Lab environment fully operational
5-6	Execute simulation scenarios; collect Security Event Logs (.evtx) and Registry hive data (.DAT) against known ground truth	Full dataset collected with ground truth documented
7	Develop Python parsing scripts; perform timestamp alignment and cross-artefact correlation using pandas	Correlation engine operational
8	Reconstruct access timelines; compute detection rate, false positive rate, and timeline completeness metrics	Timeline analysis and evaluation complete
9	Evaluate results against ground truth; compare outcomes with benchmarks from reviewed literature	Evaluation section finalised
10-11	Write all proposal sections; integrate citations and cross-references; review for coherence and word count compliance	Full draft complete
12	Final proofreading, IEEE reference formatting, ethics form completion, and Turnitin submission	Proposal submitted

7 REFERENCES

- [1] K. S. Lim, S. Y. Ooi, Y. J. Chew, and M. S. Sayeed, "Application of Machine Learning Algorithms for Anomaly Detection in Cybersecurity Threat Mitigation," *Informatica*, vol. 50, no. 9, pp. 375-382, 2026.
- [2] M. S. Mohamed and A. Arabo, "A SIEM-Integrated Cybersecurity Prototype for Insider Threat Anomaly Detection Using Enterprise Logs and Behavioural Biometrics," *Electronics*, vol. 15, no. 248, 2026, doi: 10.3390/electronics15010248.
- [3] S. U. Shaukat, S. Khan, and S. Parkinson, "Event Log Correlation for Multi-Step Attack Detection," *Security and Privacy*, vol. 9, e70151, 2026.
- [4] P. P. Sajan et al., "Analytical Study on Forensic Relevance of Windows Event Logs," *IJRASET*, vol. 13, no. 7, 2025, doi: 10.22214/ijraset.2025.72991.
- [5] Q. Do, B. Martini, J. Looi, Y. Wang, and K.-K. R. Choo, "Windows Event Forensic Process," in *Advances in Digital Forensics X, IFIP AICT 433*, pp. 87-100, 2014.
- [6] P. C. Patel et al., "A Forensic Perspective on the Use of Event Viewer for Detecting Malicious Activities and Ensuring System Integrity," *ShodhKosh*, vol. 5, no. 1, pp. 3250-3256, 2024, doi: 10.29121/shodhkosh.v5.i1.2024.5975.
- [7] J.-H. Lee and H.-Y. Kwon, "Large-Scale Digital Forensic Investigation for Windows Registry," *PLOS ONE*, 2022.
- [8] H. Studiawan, *Forensic Investigation of Event Logs by Automatic Anomaly Detection*, PhD thesis, Murdoch University, Australia, 2020.
- [9] D. Patil and A. Prabhu, "Malware Detection Through Memory Forensics and Windows Event Log Analysis," *The International Arab Journal of Information Technology*, vol. 22, no. 6, pp. 1095-1111, 2025, doi: 10.34028/iajit/22/6/5.
- [10] R. M. Achmad, D. P. Nariswari, B. A. Pratomo, and H. Studiawan, "Sysmon Event Logs for Machine Learning-Based Malware Detection," *Cyber Security and Applications*, vol. 3, p. 100110, 2025, doi: 10.1016/j.csa.2025.100110.
- [11] B. Keyogeg, M. Thompson, G. Dawson, D. Wagner, G. Johnson, and B. Elliott, "Automated Detection of Ransomware in Windows Active Directory Domain Services Using Log Analysis and Machine Learning," preprint, 2024.

- [12] S. Khan, S. Parkinson, and M. Roopak, "Reproducibility in Event-Log Research: A Parametrised Generator and Benchmark for Event-Based Signatures," arXiv preprint arXiv:2601.12978, 2026.
- [13] A. Mystakidis et al., "XAI-Driven Malware Detection from Memory Artifacts: An Alert-Driven AI Framework with TabNet and Ensemble Classification," *AI*, vol. 7, no. 66, pp. 1-33, 2026, doi: 10.3390/ai7020066.
- [14] S. Ghosh, R. K. Goyal, and K. Chowdhury, "Explainable AI-Driven Intrusion Detection System for DoS Attack Classification Using Deep Learning and Optimization Techniques," *IEEE Access*, vol. 14, pp. 5618-5645, 2026, doi: 10.1109/ACCESS.2026.3651187.
- [15] R. Vaarandi and H. Bahşi, "Using Large Language Models for Template Detection from Security Event Logs," *International Journal of Information Security*, 2025, doi: 10.1007/s10207-025-01018-y.
- [16] H. You, C. Park, D. Shin, and D. Shin, "A Comparative Performance Study of Host-Based Intrusion Detection Using TextRank-Based System Call Preprocessing and Deep Learning Models," *Electronics*, vol. 15, no. 1856, 2026, doi: 10.3390/electronics15091856.
- [17] M. Y. Amro, "Ransomware Detection: The Efficacy of Behavior-Based and Machine Learning Techniques," MSc thesis, Al-Quds Open University, Palestine, 2026.
- [18] S.-A. Drăguşin, R.-N. Boştinaru, N. Bizon, and G.-V. Iana, "AI-Based Embedded Framework for Cyber-Attack Detection Through Signal Processing and Anomaly Analysis," *Applied Sciences*, vol. 16, no. 1416, 2026, doi: 10.3390/app16031416.
- [19] M. Landauer et al., "CAM-LDS: Cyber Attack Manifestations for Automatic Interpretation of System Logs and Security Alerts," arXiv preprint arXiv:2603.04186, 2026.
- [20] P. K. Manadhata, S. Yadav, P. Rao, and W. Horne, "Detecting Malicious Domains via Graph Inference," in *Proc. ACM AISec*, Scottsdale, AZ, USA, 2014, pp. 59-60, doi: 10.1145/2666652.2666659.