iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VII Month of publication: July 2025

DOI: https://doi.org/10.22214/ijraset.2025.72991

Analytical Study on Forensic Relevance of Windows Event Logs

Dr. Priya P. Sajan[1], Archa Rajesh R[2], Jenani S[3], Milina S.S.[4], Vaishnavi Devi R[5], Vishnu Prabha P[6]
[1]*Senior Project Engineer, C-DAC, Thiruvananthapuram*
[2, 3, 4, 5, 6]*Noorul Islam Centre for Higher Education, Thuckalay*

Abstract: Event logs are crucial records automatically maintained by computer systems to track activities and operations. They serve as a key source of information for system administrators and forensic investigators to monitor, audit, and analyse both normal and suspicious activities. This research study explores the forensic relevance of Windows event logs. It further discusses the tools and techniques employed for log analysis, recovery, and centralization, emphasizing their role in enhancing organizational cybersecurity and forensic readiness.

Index terms: Event Logs, Log Integrity, Malware Detection, SIEM (Security Information and Event Management)

I. INTRODUCTION

In the evolving landscape of digital infrastructure, the integrity, security, and performance of computer systems are increasingly reliant on the effective monitoring of system activities. Event logs play a crucial role in this regard by providing a structured record of system, application, and security-related events. These logs are automatically generated by operating systems and software applications, capturing critical information such as timestamps, event sources, event IDs, and user activities.

Event logs serve as a foundational element in cybersecurity, digital forensics, and system administration. They are instrumental in identifying unauthorized access, tracing user actions, detecting anomalies, and performing post-incident forensic investigations. In Windows-based environments, the Event Viewer utility categorizes logs into several key types, including Security, System, Application, Setup, and Forwarded Event Logs. Each type serves a specific purpose in maintaining system transparency and supporting compliance with organizational audit policies.

The analysis of event logs has become an essential part of modern incident response and forensic frameworks. With the growing sophistication of cyber threats and insider attacks, organizations increasingly rely on Security Information and Event Management (SIEM) systems such as Splunk and IBM QRadar to collect, correlate, and analyse large volumes of log data. Moreover, forensic tools like FTK Imager are employed to recover deleted logs, ensuring the continuity and integrity of digital evidence.

II. RELATED WORKS

Event logs serve as a foundational component in both system administration and digital forensics. Prior research has highlighted the critical role of event log auditing in identifying system anomalies and supporting post-incident investigations. In particular, system event logs have been used to detect low-level service failures and kernel-level misbehaviors, offering early indicators of rootkits and unauthorized modifications. Application logs, which record the internal states and exceptions within specific programs, have also gained importance. Studies have shown that structured application logging significantly enhances anomaly detection by allowing correlation of software behavior with system events. Security logs, most notably those found in the Windows Security Event Log, are widely utilized for tracking authentication attempts, privilege use, and policy violations. These logs have been incorporated into Security Information and Event Management (SIEM) systems, enabling organizations to perform real-time threat correlation and risk-based alerting. Several works have explored how failed logon attempts, process creation records, and access control violations serve as key indicators of insider threats and lateral movement during cyberattacks.

System logs, though often overlooked, have proven valuable in identifying misconfigurations, startup failures, and suspicious service behavior. Research has demonstrated the utility of system log mining in failure prediction and post-mortem analysis of operational outages. Setup logs, such as setupact.log and setuperr.log, are more specific to installation and upgrade events in Windows systems. While less commonly addressed in academic literature, some forensic studies have emphasized their role in diagnosing failed upgrades, rollback conditions, and tampered installation paths. These logs are crucial when investigating cases of OS corruption or insider manipulation during system deployment phases.

Forwarded event logs have become essential in enterprise environments where logs from multiple endpoints are centralized for monitoring and correlation. Existing work has demonstrated how centralized logging improves visibility and scalability in distributed systems, allowing for efficient compliance checks and accelerated threat hunting. Despite the growing reliance on log data, relatively few studies focus on the forensic recovery of deleted or manipulated logs, particularly setup logs that may be erased to conceal unauthorized modifications. This gap in the literature motivates the present study, which aims to explore forensic techniques for recovering and analyzing critical log files in compromised or rollback-prone systems.

III. METHODOLOGY

Event logs play a critical role in monitoring and maintaining the security, performance, and stability of systems. An event log is a digital record that stores detailed information that occurrences within a system, such as user logins, software errors, security breaches, system warnings, and more.

A. Application Log

1) Structure and Functions of Application Log

To monitor system activities, user interactions, and internal application operations, software automatically creates application logs. These logs provide detailed information that helps monitor application behaviour and diagnose issues efficiently. The core functions of application logs include troubleshooting, where they help identify errors, crashes, and failures. They also support root cause analysis by tracing the sequence of events that led to a problem. In addition, logs are used for monitoring system health and application performance, as well as providing security insight by detecting unusual or suspicious behaviour. The content of application logs typically includes timestamps of events, error and warning messages, user actions and inputs, system responses, and performance metrics such as memory usage or load time. These logs are vital in cyber forensics, as they help investigators understand what happened during an incident and determine who was responsible.

2) Tools For Log Collection and Analysis

Several tools are available for collecting, managing, and analysing application logs, each offering features that support both operational monitoring and forensic investigation. Splunk is a widely used platform that provides real-time log monitoring, powerful search capabilities, and customizable dashboards. It helps detect anomalies and generate alerts based on predefined rules. The ELK Stack, consisting of Elasticsearch, Logstash, and Kibana, is another popular open-source solution. It enables users to collect and parse logs (Logstash), store and search them efficiently (Elasticsearch), and visualize the results (Kibana). Graylog is a centralized log management tool that offers similar functionality with a focus on simplicity and scalability. For forensic-specific use cases, tools like FTK Imager and Autopsy are valuable for recovering deleted log files and examining raw disk data. These tools enhance the forensic workflow by simplifying log collection, enabling timeline reconstruction, and ensuring the integrity of evidence during analysis.

3) Log Deletion and Recovery

Application logs may be removed either through automated processes or by manual intervention. Automatic deletion often occurs due to log rotation policies or limited storage, where older logs are removed after reaching a certain size or time limit. Manual deletion is more concerning, as attackers may intentionally delete logs to hide traces of their activity. In such cases, forensic investigators rely on tools like Recuva, FTK Imager, and Autopsy to recover deleted log files, provided the data has not been overwritten. Recovery is also possible through Windows Shadow Copies or system backups, if available. However, if logs have been securely erased using shredding tools or overwritten by new data, recovery becomes extremely difficult or impossible. Understanding these scenarios is essential in ensuring log preservation during forensic investigations.

4) Prevention of Log Deletion

A range of preventive measures can be employed to maintain the integrity and availability of application logs. several preventive measures can be implemented. Setting proper file permissions ensures that only authorized users or processes can access, modify, or delete logs. Centralized logging systems, such as Splunk or ELK Stack, forward logs to remote servers, making them harder to tamper with locally. Regular log backups to external drives or cloud storage help restore data in case of deletion or corruption. Implementing audit trails allows monitoring of log access and changes, providing alerts if unauthorized modifications occur. For enhanced protection, immutable storage solutions like Write Once Read Many (WORM) systems can be used, ensuring that logs cannot be altered or erased once written. These strategies collectively help preserve logs as reliable forensic evidence.

B. Security log

1) Overview of Security Logs

Operating systems maintain specialized security logs that record events pertinent to the integrity and access control of computer systems. These logs are instrumental in monitoring user behaviour, detecting unauthorized access, and identifying system changes. Within both standalone systems and networked environments, security logs track activities such as login attempts, resource access, and file deletions. The primary purpose of security logs is to aid in detecting and investigating unauthorized or suspicious activities. These logs are vital in digital forensic investigations, as they provide a chronological trail of events that can be analysed to determine the cause and scope of an incident. Security logs are frequently utilized by cybersecurity professionals, incident responders, and forensic investigators to uncover digital evidence post-incident.

2) Types of Records Stored in Security Logs

Security logs comprise various entries that facilitate the monitoring and tracing of activities within a system. These entries are essential for identifying unauthorized access, policy violations, and anomalous system behaviour. Key types of records include:

Authentication Events: Authentication events record user login and logout activities, including both successful and failed access attempts, providing critical data for security auditing and forensic investigation.

Account Management: Encompasses the creation and deletion of user accounts, password changes, privilege escalations, and modifications in group memberships.

Object Access: Tracks access to files, directories, and hardware devices, recording attempts to read, write, or delete sensitive resources.

3) Tools Used for Security Logs

Effective management of security logs requires tools capable of collecting, analysing, storing, and safeguarding log data. Prominent tools include:

Splunk: A real-time log analysis platform that supports searching, monitoring, and visualizing log data through dashboards and alerts.

IBM QRadar: A Security Information and Event Management (SIEM) solution that aggregates and analyses log data from multiple sources to detect threats.

LogRhythm: An integrated SIEM platform offering centralized log management and automated threat detection.

Auditd: The Linux auditing daemon, is responsible for monitoring system calls associated with file access and user activities, thereby maintaining detailed and comprehensive audit trails for security and forensic analysis.

4) Importance of Security Logs in Forensic Investigation

Security logs constitute a fundamental source of digital evidence, playing a pivotal role in the investigation and analysis of cyber incidents. The significance of these aspects is primarily due to the following

Digital Evidence: Provide direct records of activities performed on systems and networks.

Timeline Reconstruction: In digital forensics, timeline reconstruction facilitates the chronological sequencing of events to trace an attacker's actions.

Scope and Impact Analysis: Security logs assist in identifying the systems affected and the extent of data compromise during a security incident.

Root Cause Analysis: Help identify the vulnerability or entry point exploited during the attack.

Legal and Compliance: Support legal proceedings and ensure adherence to regulatory requirements.

5) Challenges and Recovery of Deleted Security Logs

The intricate process of recovering deleted security logs is influenced by several factors, including the deletion methodology, the operating system employed, the time elapsed since deletion, and the extent of disk overwriting. Several tools and methods are employed in recovery:

Forensic Image Acquisition: Involves creating a bit-forbit copy of the storage medium using hardware write-blockers to preserve evidence integrity.

Volume Shadow Copy (VSS): Stores file system snapshots, enabling access to previous versions of logs. Tools like Shadow Explorer, vssadmin, and FTK Imager facilitate recovery from VSS.

SIEM-Based Forwarding: Logs forwarded to SIEM platforms like Splunk, QRadar, and LogRhythm remain secure and retrievable even if deleted from the local machine.

6) Methods to Prevent Log Deletion

Implementing preventive measures is crucial to maintaining the integrity and availability of log data within secure computing environments. Common methods include:

Change File Permissions: Setting log files to read-only prevents unauthorized modification or deletion.

Secure Storage: Moving logs to protected folders, external drives, or remote servers reduces local access risks.

Cryptographic Hashing: Applying hash functions (e.g., SHA-256, MD5) to verify file integrity. Hashes can be stored separately for later validation.

Log Forwarding to SIEM: Transmitting logs to centralized servers ensures that logs are retained and protected even if local copies are compromised.

C. Setup Log

1) Introduction to Setup Logs

Setup logs are automatically generated diagnostic files that record system activity during the installation or upgrade of Windows operating systems. These logs are stored in system as %SystemDrive%\$WINDOWS.~BT\Sources\Panther\ and include files like setupact.log, which documents general actions, and setuperr.log, which captures critical errors. They offer valuable insights into system checks, hardware compatibility, software configurations, and failure points during the setup process. By providing detailed, timestamped entries, setup logs serve as a primary source of information for system administrators and forensic investigators seeking to understand upgrade behaviour and diagnose underlying issues.

2) Real-World Upgrade Failure Case Study

A practical investigation was conducted in an enterprise environment where the IT department attempted to upgrade 100 laptops from Windows 10 to Windows 11. Of these, 80 machines completed the process successfully, while 20 experienced rollbacks due to critical errors. These rollbacks occurred without providing meaningful feedback to the users, resulting in productivity loss and unstable system behaviour. Upon manual inspection, the IT team identified several key causes, including hardware incompatibility with Windows 11 requirements, insufficient disk space, system file corruption, and incomplete upgrade processes due to unexpected shutdowns. The failures required a deeper technical investigation supported by setup log analysis to resolve.

3) Role and Importance of Setup Logs in Diagnosis

Setup logs played a pivotal role in diagnosing and resolving these failures. They enabled system administrators to pinpoint the exact phase and nature of the problems. For instance, logs facilitated troubleshooting of installation failures by identifying which component or phase caused the interruption. They also assisted in diagnosing Windows Update-related issues by capturing errors in patch validation and update configuration. Moreover, the logs allowed for verification of software component setup, providing clarity on successfully installed and failed drivers. Setup logs also clarified upgrade path problems by revealing version conflicts and unsupported transitions between Windows builds. Finally, they helped detect missing or corrupted files through integrity checks and signature validation entries, proving essential in tracing sources of failure.

4) Tracing Data Within Setup Logs

To trace the underlying problems, analysts reviewed various log files, including setupact.log, setuperr.log, and BlueBox.log. The investigation focused on identifying patterns through error codes such as 0x80070070 (insufficient disk space) and 0xC1900101 (driver conflict). Timestamps were critical in reconstructing the timeline of installation and failure events. The log entries were further divided by installation phases—Down-level, SafeOS, First Boot, and Second Boot—to determine exactly when each system failed. Additionally, references to specific hardware modules and software packages enabled the team to isolate which drivers or components contributed to the failed upgrade.

5) Tools Used to Analyse Setup Logs

A variety of diagnostic tools were utilized to conduct the analysis. Microsoft’s SetupDiag was employed to automatically parse setup logs and map errors to known issues, accelerating root cause identification.

The built-in Event Viewer tool provided complementary system-level error data aligned with setup timestamps. To verify the integrity of system files before and after the upgrade, Deployment Imaging Servicing and Management Tool (DISM) and System File Checker (SFC) were used. For advanced pattern detection and multi-log querying, Log Parser Studio was used across log datasets collected from different systems. Together, these tools supported both automated and manual analysis at scale.

6) Environmental Factors Affecting Log Availability

Several environmental and system-related factors were found to affect the availability and reliability of setup logs. Automatic deletion by the operating system after rollbacks led to the loss of temporary logs. In addition, scheduled disk cleanup utilities removed important diagnostic files to free up space. In some systems, setup logs were overwritten during retry attempts, erasing critical information from the failed sessions. Disk-related errors also caused some logs to become corrupted or unreadable. In more sensitive environments, the risk of malware or insider threats tampering with or deleting log files was also noted as a potential concern, underscoring the fragility of log availability.

D. System Log

1) Importance Of System Log

The system log is essential for ensuring both the stability and security of a system’s operations**.** It detects hardware and driver issues by recording problems with components, enabling early identification and resolution of failures. It provides detailed errors and warnings that IT professionals can use to effectively troubleshoot and fix system crashes or instability. Additionally, the System Log monitors system changes, such as service start-ups, shut-downs, and driver installations, allowing administrators to track modifications over time. It supports security and compliance by helping detect suspicious system-level events and providing essential records for compliance audits. Furthermore, by analysing recurring warnings or errors, administrators can pro-actively address underlying issues, improving overall system performance and stability while reducing downtime.

2) Relevance Of System Log in Forensic Investigation

System logs are essential for forensic investigations, as they enable accurate reconstruction of incident time lines by providing precise timestamps for critical events such as system start-ups, shut-downs, and driver failures. They can offer evidence of malicious activity, with unexpected reboots, repeated driver crashes, or unusual service behaviours indicating the presence of malware, rootkits, or other attacks. Additionally, system logs corroborate security and application logs by confirming whether system-level issues coincided with suspicious user actions. Through careful analysis, investigators can attribute incidents to specific causes, distinguishing between hardware failures, software bugs, or intentional tampering. Finally, because system logs are reliable, time-stamped records, they can serve as admissible evidence in court if properly preserved, ensuring a clear chain of custody and supporting the integrity of forensic findings.

3) How Security Threats Can Affect the System Log

System logs are vulnerable to various attack techniques aimed at evading detection or obstructing forensic investigations. Attackers with elevated privileges may attempt to tamper with or delete system log entries to conceal unauthorized activities. Certain malware strains engage in log flooding by generating thousands of fake errors or warnings, overwhelming the log and making it difficult for administrator investigators to identify relevant events. Advanced threats may also disable Windows Event Log services entirely, preventing new events from being recorded and effectively blinding security monitoring tools. Although rare, sophisticated attackers can inject false entries into the system log to mislead investigators or falsely implicate legitimate processes. Even without direct tampering, the system log may contain indirect evidence of compromise, such as unexpected driver failures or repeated system restarts, which can indicate the presence of malicious activity.

4) How To Identify If the System Log is Affected

Detecting signs of tampering in system logs requires careful analysis of event patterns and log integrity indicators. Unexplained gaps in event timestamps may reveal deletion of log entries intended to hide malicious activities. Investigators should check for log clear events, such as Event ID 104 in the System Log, which indicates that the event log was cleared —a potential sign of tampering if performed without authorization. Correlating the System Log with Security and Application logs can help determine whether there was activity during periods missing from the system events, highlighting inconsistencies.

Additionally, an unusual reduction in log file size may suggest that logs were cleared or replaced. Events indicating that the Windows Event Log service stopped unexpectedly can also signal attempts by attackers to disable logging and avoid detection.

5) *Tools To Analyse the System Log*

Several tools are available to analyze and manage system logs effectively. SolarWinds Event Log Analyzer offers centralized log collection, real-time monitoring, and automated alerts to help detect suspicious activities across networks. Windows Event Viewer is a native Windows tool that enables administrators to access and analyse system, security, and application logs on both local and remote devices. Log Parser Studio provides a powerful GUI for running SQL-like queries against Windows logs and other structured text files, enabling advanced log analysis. Graylog is an open-source platform designed for centralized log management and analysis, featuring customizable dashboards and search capabilities. Splunk is a widely used commercial solution that collects, indexes, and analyzes large volumes of machine-generated data, offering advanced visualization and alerting features. ELK Stack, which consists of Elasticsearch, Logstash, and Kibana, is an open-source suite for ingesting, storing, and visualizing logs, widely adopted for scalable log analysis and security monitoring.

6) *Prevention Methods of System Log*

To secure system logs from tampering and ensure their availability for forensic investigations, several best practices should be implemented. Restrict access permissions by configuring strict file and folder permissions so only authorized accounts can read or modify log files. Enable audit policies to record key events, such as login attempts or object access, enhancing visibility into suspicious activities. Forward logs to a central server to reduce the risk of local tampering by storing copies off-host. Use file integrity monitoring tools to detect unauthorized changes to log files by alerting administrators to modifications. Schedule regular log backups to preserve historical records and enable recovery if logs are corrupted or deleted. Keep systems patched and updated to mitigate vulnerabilities that attackers could exploit to gain unauthorized access. Limit administrator access by enforcing the principle of least privilege and requiring strong authentication for privileged accounts. Monitor for log clear events, such as Event ID 104 in Windows, which can signal attempts to cover malicious activity by erasing evidence.

E. Forwarded Event

1) *Purpose of Using Forwarded Event Logs*

Forwarded event logs are used to centralize event data from multiple systems into a single location. This centralization allows administrators to monitor system activities more efficiently across the network. By having all logs in one place, it becomes easier to detect problems or potential security threats quickly. Additionally, it saves time and effort for system administrators by streamlining the troubleshooting and maintenance processes, thereby improving overall administrative efficiency.

2) *Mechanism of Operation*

In a Windows Event Forwarding (WEF) setup, source computers on the network are configured to send their event logs to a designated collector computer. This collector is set up to receive and store these logs centrally. This uninterrupted methodology facilitates log data capture with no requirement for manual input.

3) *Tools Used in Forwarded Events*

Several tools are involved in managing and analysing forwarded event logs. The Event Viewer provides a graphical user interface (GUI) for viewing logs on the collector system. PowerShell is used for automating tasks and managing logs through scripts. Wevtutil is a command-line utility that allows detailed control over event logs, such as querying and exporting. Windows Event Collector (WEC) is the service responsible for receiving forwarded logs, while Windows Event Forwarding (WEF) is the core technology that enables the transmission of logs from source to collector machines. Additionally, third-party tools like Event Log Analyzer (e.g., ManageEngine) serve as Security Information and Event Management (SIEM) platforms, offering advanced analysis, alerting, and reporting capabilities.

4) *Benefits of Forwarded Events*

Forwarded event logging provides significant advantages for organizations, starting with the centralized collection of logs from multiple systems into one location. This consolidation simplifies log analysis and improves visibility across the network. It also enhances troubleshooting efficiency, as administrators can quickly identify and resolve issues by reviewing logs from various sources in one place. Furthermore, the system is highly scalable, making it suitable for large network environments with many endpoints, as it can be configured to handle a wide range of source machines and logging requirements.

5) Disadvantages of Forwarded Events

Despite its benefits, forwarded event logging also comes with certain drawbacks. One major challenge is the complexity of the configuration process, which involves setting up source and collector machines, subscriptions, and possibly Group Policy Objects (GPOs). There is also a risk of misconfiguration, which could lead to gaps in log collection or exposure of sensitive data. Additionally, troubleshooting issues related to the collector—such as overloads, connection failures, or incorrect settings—can be difficult and time-consuming, particularly in large or dynamic environments.

6) Deletion of Forwarded Event Logs

Yes, forwarded event logs can be deleted, but only by users with administrative privileges. From a forensic perspective, the deletion of these logs is highly suspicious and may indicate malicious activity, such as privilege escalation, malware execution, or attempts to hide unauthorized actions. Such deletions are often considered anti-forensic behaviour, making them a red flag during digital investigations.

7) Prevention of Deleting Forwarded Event Logs

To prevent deletion of forwarded logs, limit admin access, enable audit log clearance, and ensure regular backups. For an enhanced security posture, logs should be stored on Write Once, Read Many (WORM) media to achieve immutability, and Security Information and Event Management (SIEM) systems should be deployed for unified and proactive threat monitoring.

8) Recovery of Deleted Forwarded Event Logs

Deleted forwarded logs can sometimes be recovered using forensic tools like EnCase or FTK. If regular backups were configured, logs may also be restored from those archives. Additionally, if logs were already forwarded to a SIEM platform before deletion, they can be retrieved from the SIEM's storage.

IV. RESULT AND DISCUSSION

The analysis presented in this study underscores the critical role that event logs play in modern digital forensics and cybersecurity operations. Through the detailed examination of five major event log types—Application, Security, System, Setup, and Forwarded Event Logs—this work has highlighted their respective structures, forensic relevance, and the tools utilized for their collection, analysis, and preservation.

The Application Logs demonstrated their value in tracking software behaviour and system anomalies, providing rich contextual data for identifying crash patterns and suspicious application activity. The methodology applied using tools such as Splunk, ELK Stack, and FTK Imager enabled effective monitoring and evidence retrieval, even in cases involving deleted or tampered logs.

Security Logs, central to forensic investigations, were shown to be essential for authentication tracking, account management, and access control validation. Their integration with SIEM platforms such as IBM QRadar and LogRhythm significantly enhances real-time detection of security breaches and supports audit compliance. Challenges associated with their deletion were addressed through Volume Shadow Copy recovery and centralized logging approaches.

The study of Setup Logs, often overlooked in traditional investigations, revealed their importance in diagnosing failed system upgrades and tracing installation errors. Tools such as SetupDiag, DISM, and Log Parser Studio proved vital in reconstructing installation timelines and identifying root causes of failures, particularly in enterprise-scale rollouts.

System Logs provided evidence of low-level operations and hardware interactions. The presence of tampering or log manipulation could be identified through Event ID analysis, file integrity verification, and correlation with other logs. Prevention strategies such as restricting administrative privileges, enabling audit policies, and deploying immutable storage were found to be effective.

Finally, Forwarded Event Logs were shown to be crucial for centralized monitoring across distributed environments. Although offering scalability and efficiency, their reliance on correct configuration and potential vulnerability to administrative misuse were acknowledged. Preventive mechanisms such as access controls, regular backups, and WORM-based storage were recommended to ensure their forensic value.

V. CONCLUSION

Event logs represent a critical component in digital forensic investigations and cybersecurity operations. In this study, five primary categories of Windows event logs—Application, Security, System, Setup, and Forwarded Event Logs—were examined for their roles in system monitoring, threat detection, and forensic analysis. Each log type was found to provide distinct insights into user behaviour, system processes, and application performance, thereby supporting incident reconstruction and audit compliance.

The methodologies employed involved the use of industry-standard tools, including Splunk, FTK Imager, SetupDiag, and Event Viewer, to collect, analyse, and interpret log data. Techniques for detecting log tampering, recovering deleted logs, and preventing unauthorized access were also implemented. It was observed that centralized log collection and the integration of Security Information and Event Management (SIEM) systems significantly enhanced forensic readiness and data integrity.

It is concluded that the implementation of secure logging practices, coupled with reliable recovery mechanisms, is essential for maintaining the evidentiary value of event logs. Future research may focus on the adoption of machine learning algorithms for automated anomaly detection and the deployment of tamper-evident or immutable logging architectures to further strengthen cyber defence and forensic capabilities.

REFERENCES

[1] Y. Gao, H. Kim, and R. Buyya, "Anomaly detection using system logs: A survey," IEEE Transactions on Services Computing, vol. 14, no. 1, pp. 1–19, Jan.–Feb. 2021, doi: 10.1109/TSC.2019.2905587.

[2] H. Duan, Y. Zhang, and X. Li, "Security event log analysis using deep learning," IEEE Access, vol. 8,pp.120885–120895,2020,doi: 10.1109/ACCESS.2020.3005798.

[3] A. Khatuya and B. Mishra, "Application log analysis for anomaly detection in microservices architecture," in Proc. IEEE Int. Conf. on Big Data (BigData), Atlanta, GA, USA, 2020,pp.2697–2704,doi: 10.1109/BigData50022.2020.9378371.

[4] J. Stearley and A. Oliner, "Bad words: Finding faults in spirit's syslogs," in Proc. IEEE Int. Conf. on Dependable Systems and Networks(DSN), Edinburgh,U.K.,2007,pp.218–227doi: 10.1109/DSN.2007.56.

[5] R. Ahmed and V. Das, "A forensic approach to analysing Windows setup logs," International Journal of Cyber Forensics, vol. 5, no. 3, pp. 45–58, 2022.

[6] N. Tanaka, M. Harwood, and T. Erickson, "Forwarded event logs for distributed monitoring and forensics," in Proc. IEEE Int. Conf. on Cloud Engineering (IC2E), Orlando, FL, USA, 2018, pp. 190–197, doi: 10.1109/IC2E.2018.00041.

[7] D. Becker and L. Wang, "System log mining for early failure prediction," in Proc. IEEE Int. Conf. on Cloud Computing, San Francisco, CA, USA, 2020, pp.55–62, doi: 10.1109/CLOUD49709.2020.00016.

[8] R. McMillen, "Investigating Windows 10 setup logs using SetupDiag," in Proc. 13th Int. Conf. on Digital Forensics & Cyber Crime (ICDF2C), 2021.

[9] S. Rao and H. Kim, "Insider threat detection using security event logs and behavioral analytics," IEEE Access, vol. 9, pp. 144232–144245, 2021, doi: 10.1109/ACCESS.2021.3120846.

[10] N. Hubballi and V. Suryanarayanan, "False alarm minimization techniques in signature-based intrusion detection systems: A survey," Computer Communications, vol. 49, pp. 1–17, 2014, doi: 10.1016/j.comcom.2014.04.008.