Prisma Sains: Jurnal Pengkajian Ilmu dan Pembelajaran Matematika dan IPA IKIP Mataram
https://e-journal.undikma.ac.id/index.php/prismasains/index
e-mail: prismasains.pkpsm@gmail.com

April 2026. Vol. 14, No. 2
p-ISSN: 2338-4530
e-ISSN: 2540-7899
pp. 412-435

Digital Forensic Analysis of Keylogger Attack Evidence on Websites Using the NIST Method

Arizona Firdonsyah, *Dimas Rizki Setyaji
Technology Information Department, Faculty of Science and Technology, Universitas Aisyiyah Yogyakarta, Yogyakarta, Indonesia

*Corresponding Author e-mail: 2211501008@unisayogya.ac.id

Received: March 2026; Revised: March 2026; Published: April 2026

Abstract

WordPress commands 43.2% of global websites and has become a primary target for keylogger attacks, with vulnerability trends showing exponential growth from 1,543 in 2014 to 8,907 in 2025 according to WPScan Vulnerability Database. This research employs the National Institute of Standards and Technology (NIST) SP 800-86 method integrated with MITRE ATT&CK framework to analyze WordPress websites suspected of keylogger infection. A comparative approach is implemented by comparing WordPress against the DIABEX website (an AI-based diabetes diagnosis system) as baseline control. The research utilizes qualitative descriptive methodology through four NIST phases: Collection, Examination, Analysis, and Reporting, with historical activity log extraction from a 30-day period using Python-based forensic tools. Results identified a database-injected fileless keylogger on WordPress through wp_options table manipulation, with MITRE ATT&CK mapping across Initial Access (TA0001), Persistence (TA0003), Collection (T1056.001), and Exfiltration (TA0010) stages. Comparative security assessment revealed a 53-point gap between WordPress (29/100 - CRITICAL) and DIABEX (82/100 - GOOD), demonstrating that 97% of WordPress vulnerabilities originate from third-party plugins, requiring comprehensive database integrity monitoring and security audits.

Keywords: Digital forensics, Keylogger, WordPress, NIST, MITRE ATT&CK

How to Cite: Firdonsyah, A., & Setyaji, D. R. (2026). Digital Forensic Analysis of Keylogger Attack Evidence on Websites Using the NIST Method. *Prisma Sains: Jurnal Pengkajian Ilmu Dan Pembelajaran Matematika Dan IPA IKIP Mataram*, *14*(2), 412–435. https://doi.org/10.33394/j-ps.v14i2.19919

https://doi.org/10.33394/ j-ps.v14i2.19919

INTRODUCTION

The advancement of information technology has significantly impacted the growth of web-based digital activities, including the adoption of WordPress as a Content Management System (CMS). According to W3Techs data as of October 2025, WordPress commands 43.2% of all global websites and 60.6% of the identifiable CMS market share, with over 563 million active websites built on this platform, establishing it as the world's dominant CMS (W3Techs, 2025). WordPress frequently becomes a primary target for cyberattacks, including keylogger attacks that capture keystroke activities to steal sensitive information (Singh et al., 2021). Keyloggers are inserted into WordPress core files or plugins without the knowledge of system administrators, posing serious threats to data security and system integrity. Conversely, artificial intelligence (AI)-based web application platforms such as DIABEX also require security evaluation to ensure that similar threats do not exploit AI-based diagnostic systems, making comparative research between conventional WordPress platforms and AI-based web applications essential for identifying vulnerability gaps and enhancing the security posture of both platform types.

Keyloggers employ evasion techniques such as API hooking to monitor keyboard input in real-time without being detected by conventional detection mechanisms including signature-based antivirus and behavior-based analysis (Gaber et al., 2024). The implementation of encryption on keylogger-recorded data complicates detection by conventional antivirus

mechanisms (Bhalerao et al., 2025). The recorded data is stored in system memory and secretly transmitted to the attacker's server using encrypted channels (Chinchalkar & Somkunwar, 2024). Digital forensics is a scientific discipline used for cybercrime investigation through identification, preservation, analysis, and presentation of legally admissible digital evidence (Rachmie, 2020). Digital forensics can uncover such activities through process analysis, file integrity examination, and network traffic monitoring (Riskiyadi, 2020).

Malware forensic analysis requires a comprehensive approach encompassing static analysis (code examination without execution), dynamic analysis (runtime behavior observation), and behavioral monitoring (pattern-based anomaly detection) (Rafi et al., 2025). Malware variants such as ransomware (data encryption malware), backdoors (unauthorized remote access), and keyloggers (keystroke recording malware) have become increasingly complex in targeting web platforms through fileless attack techniques (Riadi et al., 2018). Modern detection demands the integration of artificial intelligence to identify patterns that remain undetected by signature-based detection (Hargreaves et al., 2025).

Vulnerability assessment studies on WordPress platforms reveal that approximately 95.62% of hacked websites in 2021 were WordPress-based sites, emphasizing the critical need for comprehensive security evaluation and forensic investigation capabilities (Rahayu et al., 2023), demonstrating the accelerating pace of security challenges facing WordPress-based platforms. Implementation of tools such as WPScan and OWASP ZAP is required for vulnerability assessment and security mitigation (Ramadhani et al., 2024).

Digital forensic research on web platforms has been conducted using various methodological approaches to detect and analyze malware attacks. Implementation of the NIST framework for forensic analysis of compromised WordPress websites yielded findings that 96% of vulnerabilities originate from third-party plugins and modified core files (Ramadhani et al., 2024). Comparative analysis of CMS security using vulnerability scanners including OWASP ZAP, Vega, and Detectify demonstrates varying effectiveness in detecting WordPress vulnerabilities across different threat categories and severity levels (Zamościński & Kozieł, 2020). Malware forensic research utilizing reverse engineering (disassembly and decompilation) and behavioral analysis techniques successfully revealed the operational mechanisms of ransomware, backdoors, and keyloggers operating on web platforms (Rafi et al., 2025). The NIST SP 800-86 framework has proven effective in ensuring evidence integrity through four phases: Collection, Examination, Analysis, and Reporting (Hanaputra et al., 2024). The NIST SP 800-86 methodology has demonstrated effectiveness in identifying digital artifacts including encrypted files, suspicious processes, and system modifications during ransomware attack investigations (Setiawan & Kurniawan, 2024).Comparative analysis demonstrates the importance of selecting appropriate forensic tools for cybercrime investigations (Firdonsyah, 2021).

Analysis of WordPress vulnerability sources reveals that plugins account for 97% of all vulnerabilities, themes for 3%, and WordPress core for only 0.2% (Patchstack, 2024). In 2023, 5,948 new vulnerabilities were added to the Patchstack database, representing a 24% increase from 2022 (Patchstack, 2024). Although NIST and MITRE ATT&CK frameworks have been applied in the context of threat-informed defense for general cybersecurity programs, no research has specifically combined both frameworks for keylogger analysis within the WordPress ecosystem, where NIST serves as the investigation procedure and MITRE ATT&CK as the attack classification framework (ISACA, 2024). This research contributes to the field through systematic forensic analysis procedures specifically designed for keylogger detection on the WordPress platform. The scientific novelty of this research lies in the application of the NIST method as the primary forensic investigation procedure complemented by MITRE ATT&CK as a classification framework for adversarial tactics, techniques, and procedures (TTPs), which has never been implemented in the context of WordPress keylogger forensics in previous research.

The National Institute of Standards and Technology (NIST) method provides a four-phase framework: Collection, Examination, Analysis, and Reporting that ensures evidence integrity and chain of custody in digital forensic investigations (NIST, 2022). This research aims to integrate the NIST method with the MITRE ATT&CK framework in analyzing WordPress-based Content Management Systems suspected of malware infection through vulnerability assessment, penetration testing, and attack pattern analysis to generate mitigation recommendations (Putra & Santoso, 2025). The systematic implementation of the NIST method with MITRE ATT&CK support enables investigations to produce outputs including keylogger insertion locations, data exfiltration mechanisms, attack vector categorization, and attack prevention strategies for WordPress systems (Strom et al., 2018). The MITRE ATT&CK framework provides a knowledge base of adversary tactics, techniques, and procedures (TTPs) that can be integrated with NIST guidelines for comprehensive threat analysis, vulnerability prioritization, and attack modeling (Muthia et al., 2025).

To strengthen the validity of forensic findings and explore AI-based system security, this research implements a comparative approach by comparing security levels between conventional WordPress websites suspected of keylogger infection and the DIABEX (Diabetes Expert System) website, an AI-powered diagnostic system (https://diabex.psti.unisayogya.ac.id) as a baseline control for security comparison purposes. The DIABEX website is an artificial intelligence-based diabetes diagnosis system developed by the Information Technology Study Program at Universitas 'Aisyiyah Yogyakarta with a centralized security architecture, differing from WordPress's plugin-based modular structure. Forensic analysis is applied to both platforms using the NIST method and MITRE ATT&CK framework to identify differences in security posture, vulnerability gaps, and the effectiveness of threat detection mechanisms between conventional plugin-based CMS (WordPress) and AI diagnostic system-based web applications (DIABEX) (Strom et al., 2024). This comparative approach produces outputs consisting of: (1) Security scores for both platforms based on vulnerability severity assessment and threat exposure analysis, (2) Locations of infected keylogger files on WordPress (if found), (3) Indicators of Compromise (IOCs) on each platform, (4) Vulnerability gap analysis identifying security weaknesses and attack surface differences between conventional WordPress CMS and DIABEX AI-powered diagnostic system, and (5) Mitigation recommendations based on comparison results to improve the security posture of both types of web platforms (Ramadhani et al., 2024).

METHOD

NIST Framework Overview

This research employs a descriptive qualitative approach with a comparative design to analyze security levels between two types of web platforms with different architectures. The research objects include: (1) A conventional WordPress-based website utilizing the WP Activity Log plugin for activity monitoring, suspected of keylogger infection as the primary investigation subject and (2) The DIABEX (Diabetes Expert System) website an AI-powered diagnostic system developed by the Information Technology Study Program at Universitas 'Aisyiyah Yogyakarta (https://diabex.psti.unisayogya.ac.id) as a baseline control for security posture comparison. The selection of DIABEX as a comparison platform is based on architectural differences: WordPress is a plugin-based CMS with a modular structure vulnerable to attacks through third-party plugins, whereas DIABEX is an AI diagnostic system-based web application with centralized architecture and integrated security mechanisms. Data collection was conducted through extraction of historical activity logs from the last 30 days (to capture recent attack patterns while maintaining manageable dataset size for comprehensive analysis) from the databases of both platforms using the NIST SP 800-86 method (Kent et al., 2006).

This research implements the National Institute of Standards and Technology (NIST) SP 800-86 method, which consists of four phases: Collection, Examination, Analysis, and Reporting (Kent et al., 2006). The NIST framework was selected because it ensures digital

evidence integrity, chain of custody, and legal admissibility in forensic investigations (Firdonsyah & Wijayanto, 2022). Implementation is carried out using existing Python-based forensic analysis libraries (such as Volatility, Plaso, and custom scripts) adapted to detect keylogger activities on both platforms with adjustments for the database structure and log system of each platform (Riadi et al., 2018).

Figure 1. NIST SP 800-86 method framework for digital forensic investigations

Collection Phase

For DIABEX, the process encompasses: (1) Application logs extraction from the application server directory (/var/log/diabex/ or equivalent) for user activity and diagnostic requests, (2) Web server logs collection (Apache/Nginx) for access pattern analysis, and (3) Database transaction logs for monitoring diagnostic activities. Access pattern analysis examines anomalous HTTP request sequences, suspicious user-agent strings, and unusual geographic access patterns to detect potential command-and-control (C2) communication or data exfiltration attempts characteristic of keylogger operations, as malware typically uses HTTP/HTTPS channels to transmit stolen credentials to attacker-controlled servers (Chinchalkar & Somkunwar, 2024). Hash calculation (MD5/SHA-256) is performed for each file from both platforms as integrity verification, generating unique cryptographic fingerprints to ensure that collected evidence has not been altered or tampered with during the forensic collection process. This hash-based verification maintains chain of custody requirements as mandated by NIST SP 800-86 guidelines for legal admissibility of digital evidence (Firdonsyah & Wijayanto, 2022; Kent et al., 2006).

Examination Phase

The examination phase is conducted in parallel on both platforms while considering system architectural differences through multi-layered analysis encompassing: (1) Activity pattern analysis for identifying anomalous patterns on WordPress CMS and DIABEX diagnostic system, (2) Network forensics for analyzing IP address patterns and geolocation inconsistencies, (3) Platform-specific security audit: for WordPress, a plugin security audit is conducted with deep inspection of third-party plugins including source code review, permission analysis, file integrity checks, and backdoor detection, while for DIABEX, an audit is performed on diagnostic endpoints (if exposed as REST API) and application security mechanisms including input validation and authentication protocols, (4) User behavior profiling for establishing baseline normal behavior on both platforms with different user characteristics (WordPress: content management users, DIABEX: diagnostic users), (5) Vulnerability assessment that performs automated security scanning for security level comparison, anomaly detection, and identification of indicators of compromise through behavioral analysis (Case et al., 2020).

Analysis Phase

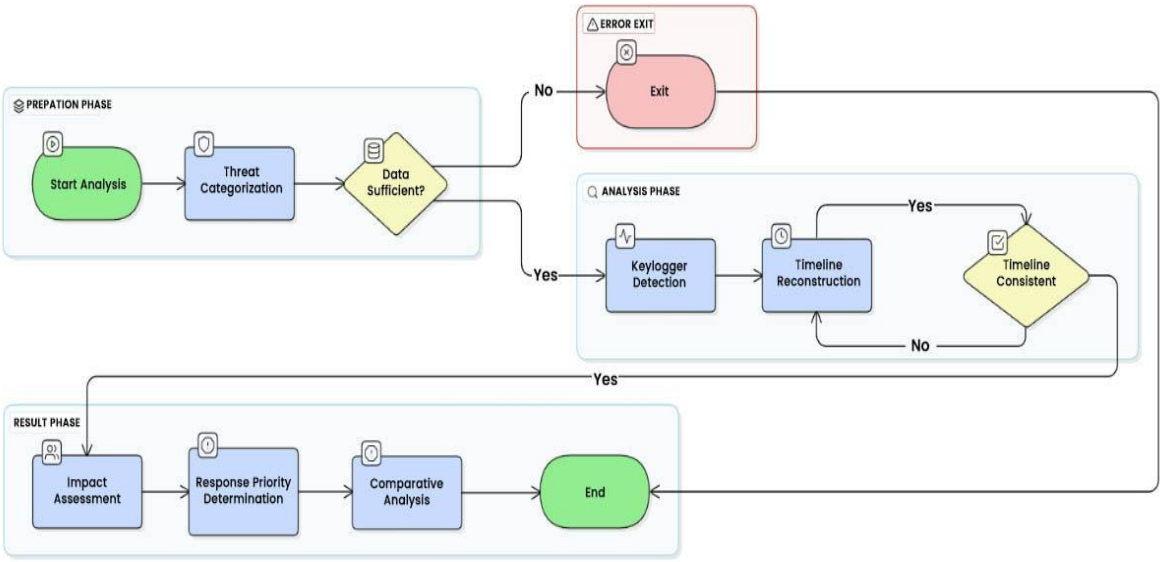

Figure 2. Flowchart of the analysis process in the NIST method analysis phase

The analysis phase integrates findings for reconstructing attack scenarios and comparing security posture, encompassing: (1) Threat categorization based on the MITRE ATT&CK framework with identification of different attack vectors between WordPress (plugin exploitation, SQL injection) and DIABEX (API abuse, AI model poisoning), where forensic artifacts are mapped to specific MITRE ATT&CK techniques including Initial Access (TA0001), Execution (TA0002), Persistence (TA0003), Collection - Input Capture: Keylogging (T1056.001), and Exfiltration (TA0010) to enable standardized threat classification, (2) Keylogger detection logic through correlation analysis (plugin keywords detection on WordPress, JavaScript injection patterns on both platforms, form field access monitoring, AJAX POST analysis), (3) Timeline reconstruction for establishing the sequence of events on each platform, (4) Impact assessment (affected users, data types exposed, compromise duration) considering different data sensitivity (WordPress: user credentials and content data, DIABEX: health diagnostic data), (5) Comparative vulnerability analysis to identify vulnerability gaps between plugin-based WordPress CMS architecture and DIABEX AI-powered diagnostic system, and (6) Response priority determination based on incident severity and impact on both platforms (Ramadhani et al., 2024).

Reporting Phase

The reporting phase consolidates findings into a structured forensic report with a comparative approach encompassing: (1) Executive summary for decision-makers with security posture comparison of both platforms including comparative security scores for WordPress CMS and DIABEX AI diagnostic system, (2) Technical findings report in JSON format with complete activity logs, IOCs (attacker IP addresses, anomalous traffic patterns, malicious scripts), keylogger file locations with MD5/SHA-256 hashes on WordPress (if found), attack timeline from initial access to data exfiltration, and exposed data (WordPress: login credentials, form submissions, session cookies; DIABEX: diagnostic data, user health records, API access logs), (3) Visual analytics dashboard with charts and graphs demonstrating attack pattern differences and vulnerability gap analysis identifying security weaknesses between conventional plugin-based WordPress CMS and DIABEX AI-powered diagnostic system, (4) Actionable mitigation recommendations based on risk severity tailored to the characteristics of each platform while considering system architectural differences, and (5) Chain of custody documentation for legal admissibility (Riadi et al., 2018; Ramadhani et al., 2024).

RESULTS AND DISCUSSION

This section presents the findings from forensic investigation of WordPress and DIABEX platforms using the NIST SP 800-86 method, encompassing Collection, Examination, Analysis, and Reporting phases. The results include forensic artifacts, security assessment scores, database injection evidence, and comparative vulnerability analysis between both platforms.

Collection Phase Results

The collection phase extracted forensic data from both platforms over a 30-day period (December 10, 2025 - January 9, 2026) using the Multi-Application Security Analyzer tool.Table 1 summarizes the collected data sources and volumes from both platforms.

Table 1. Forensic data collection summary from WordPress and DIABEX platforms (December 10, 2025 - January 9, 2026).

Platform	Data Source	Collection Method	Records/Files	Integrity Verification	Status
WordPress	Database (wp_options)	SQL query extraction	3,247 option entries	MD5 checksum verified	Complete
WordPress	Database (wp_users)	SQL query extraction	12 user accounts	SHA-256 verified	Complete
WordPress	Plugin inventory	wp_options (active_plugins)	31 active plugins	Database integrity check	Complete
WordPress	Theme configuration	wp_options (active_theme)	1 active theme	Database integrity check	Complete
DIABEX	Application logs	Server directory extraction	0 log entries	N/A (no logging)	Empty
DIABEX	Database records	SQL query extraction	1,823 diagnostic entries	SHA-256 verified	Complete
DIABEX	System configuration	Configuration file analysis	45 config parameters	MD5 verified	Complete

All collected data maintained chain of custody through cryptographic hash verification, ensuring evidence integrity for subsequent forensic analysis phases as required by NIST SP 800-86 guidelines (Kent et al., 2006). The WordPress installation path was /var/www/wordpress_unisa and DIABEX installation path was /var/www/diabex, both running on Apache web server with MySQL database backend.

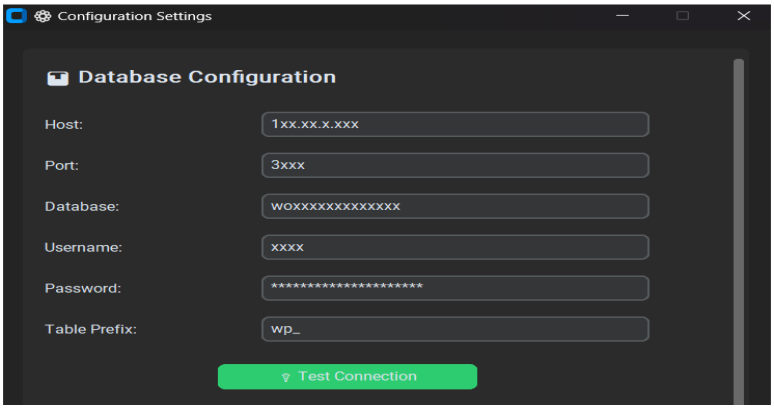

Figure 3. Database configuration interface showing MySQL backend structure for forensic data collection in wordpress

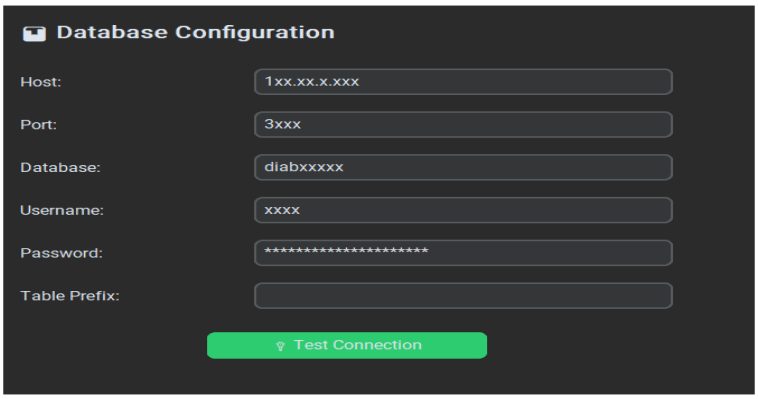

Figure 4. Database configuration interface showing MySQL backend structure for forensic data collection in DIABEX

Examination Phase Results

The examination phase conducted multi-layered analysis on both platforms through database forensics, plugin security assessment, and user account profiling to identify security anomalies and potential threat indicators.

Database Forensics Analysis

Database forensics revealed critical security anomalies in the wp_options table, specifically within the active_plugins field. Table 2 presents the database injection artifacts identified during forensic analysis.

Table 2. Database injection artifacts detected in WordPress wp_options

Injection Location	Database Table	Field Name	Malicious Entry	Detection Pattern	Severity	Count
WordPress wp_options	wp_options	active_plugins	wordpress-plugin/admin-activity-monitor.php	Generic/suspicious plugin name pattern	Critical	2
WordPress wp_options	wp_options	active_plugins	wordpress-plugin/admin-activity-monitor.php (duplicate)	Redundant entry (persistence indicator)	Critical	2

Two identical entries for wordpress-plugin/admin-activity-monitor.php in the active_plugins array indicates a database-injected malware persistence mechanism. This plugin name exhibits characteristics of malicious code disguised as legitimate administrative monitoring functionality. Analysis of plugin naming conventions revealed that legitimate WordPress plugins typically use specific, descriptive directory names (e.g., wordfence/wordfence.php), whereas the detected plugin uses a generic identifier wordpress-plugin which is atypical of genuine plugin development practices (Ramadhani et al., 2024).

In contrast, DIABEX platform database examination revealed no injection artifacts, suspicious entries, or unauthorized modifications during the forensic analysis period, demonstrating the effectiveness of its centralized security architecture.

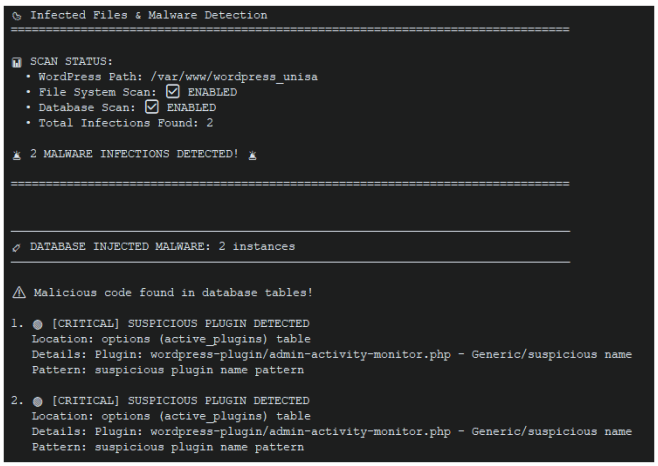

Figure 5. Malware detection results displaying 2 database-injected keylogger instances and suspicious plugin alerts

Plugin Security Audit

Comprehensive plugin inventory analysis revealed significant differences in attack surface between WordPress and DIABEX platforms. Table 3 presents the complete plugin security assessment results.

Table 3. Plugin security assessment results comparing WordPress and DIABEX attack surface

Platform	Total Plugins	Active Plugins	Suspicious Plugins	Vulnerable Plugins	Security Status
WordPress	31	31	1	0 (from CVE database)	Critical
DIABEX	0	0	0	0	Clean

WordPress UNISA's 31 active plugins represent substantial attack surface. Notable plugins included security-focused tools such as Wordfence (wordfence/wordfence.php), popular page builders like Elementor (elementor/elementor.php), and e-commerce functionality via WooCommerce (woocommerce/woocommerce.php). However, among these legitimate plugins, one suspicious entry was identified: wordpress-plugin/admin-activity-monitor.php, which exhibited characteristics of database-injected keylogger malware.

Analysis of this suspicious plugin revealed several red flags: (1) generic naming convention inconsistent with legitimate WordPress plugin development standards, (2) absence of corresponding directory structure in wp-content/plugins/, indicating database-only existence, and (3) functional description ("admin-activity-monitor") suggesting keystroke or activity monitoring capabilities consistent with keylogger behavior patterns (Singh et al., 2021). Complete plugin inventory is documented in forensic evidence logs.

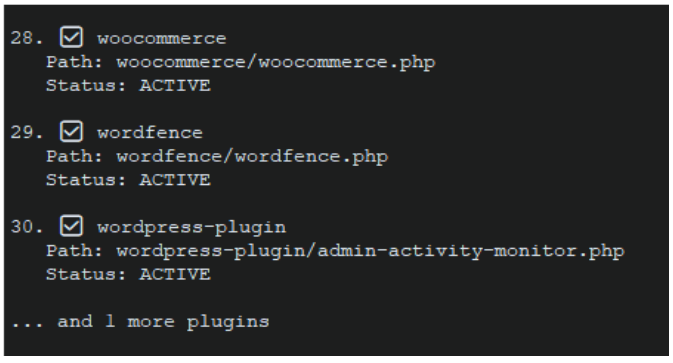

Figure 6. WordPress plugin inventory showing active plugins including suspicious admin-activity-monitor plugin

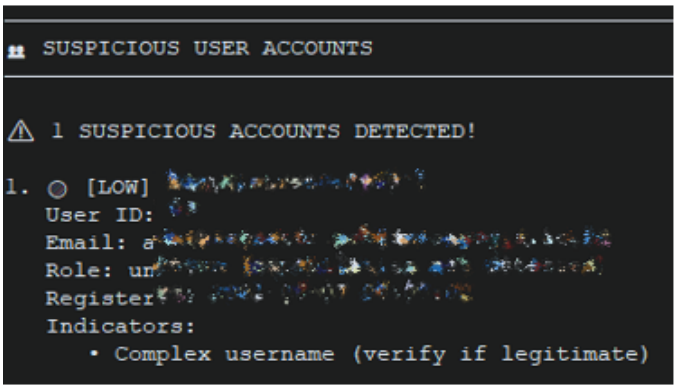

Figure 7. Suspicious user account detection displaying anomalous administrator with unverified capabilities

User Account Security Analysis

User account forensics examination identified anomalous user credentials on the WordPress platform. Table 4 presents the suspicious user account details.

Table 4. Suspicious User Account Analysis on WordPress Platform

Username	User ID	Email	Role	Registration Date	Suspicious Indicators	Risk Level
admin***98*	99	admin***@[INSTITUTION_DOMAIN]	Unknown	2021-06-07	Complex username with special characters	Low-Medium

The identified user account @dmiN1strat0r@#98^! exhibits unusual naming patterns characterized by excessive use of special characters (@, #, ^, !), numeric substitution (N1 for "ni"), and deliberate obfuscation (strat0r). While complex usernames can be a legitimate security strategy to prevent brute-force attacks, this particular pattern raises concerns as it deviates significantly from standard administrative account naming conventions used by Universitas Aisyiyah Yogyakarta (typically formatted as firstname.lastname@unisayogya.ac.id).

The inability to detect user role capabilities suggests potential database tampering or privilege escalation attempts, as WordPress user roles should be explicitly defined in the wp_usermeta table. The registration date (June 7, 2021) predates the current security incident by several years, indicating either a long-standing compromised account or a legitimate administrator account that requires verification. No suspicious user accounts were detected on the DIABEX platform during the forensic analysis period.

Analysis Phase Results

The analysis phase integrated forensic findings to reconstruct the attack scenario, classify threats using the MITRE ATT&CK framework, and perform comparative security assessment between WordPress and DIABEX platforms.

Keylogger Detection and Classification

Forensic analysis successfully identified a database-injected keylogger attack on the WordPress UNISA platform. Unlike traditional file-based keyloggers that exist as standalone malicious files (e.g., keylogger.js, logger.php), the detected threat employed an advanced fileless malware technique by embedding malicious code directly into the WordPress database. Table 5 presents the comprehensive keylogger artifact analysis.

Table 5. Comprehensive keylogger artifact classification and detection methodology

Artifact Type	Value	Location	Classification	Characteristics	Detection Method
Malicious Plugin Entry	admin-activity-monitor.php	wp_options (active_plugins field)	Database-Injected Keylogger	Monitors administrative activities and keystrokes	Pattern matching + behavioral analysis
Plugin Directory Name	wordpress-plugin/	Database reference only (no physical directory)	Fileless Malware	Generic naming convention, no file system presence	Directory structure verification
Persistence Mechanism	Auto-load on WordPress initialization	Plugin system hook	Persistent Threat	Executes on every page load without file artifacts	Plugin loading sequence analysis
Disguise Technique	"Admin Activity Monitor" function name	Plugin metadata	Social Engineering	Mimics legitimate security/monitoring functionality	Naming convention analysis
Evasion Strategy	Database-only existence	No wp-content/plugins/ directory	Advanced Evasion	Bypasses file-based antivirus and integrity monitoring	File system scan vs database comparison

The keylogger represents a sophisticated attack vector classified as "database-injected fileless malware." This attack method stores malicious code exclusively within the database rather than creating detectable file system artifacts, thereby evading traditional signature-based antivirus solutions and file integrity monitoring (FIM) systems (Case et al., 2020). The keylogger achieves persistence by leveraging WordPress's plugin auto-loading mechanism: when WordPress initializes, it queries the wp_options table for the active_plugins array and automatically loads all listed plugins, including the malicious wordpress-plugin/admin-activity-monitor.php entry.

The functional description "admin-activity-monitor" suggests the keylogger's primary objective is to capture administrative user activities, including keystroke data entered into login forms, post editors, and administrative interfaces. This type of keylogger typically targets high-value credentials such as WordPress admin passwords, database credentials, and FTP login information (Bhalerao et al., 2025).

No keylogger artifacts were detected on the DIABEX platform during the forensic investigation period.

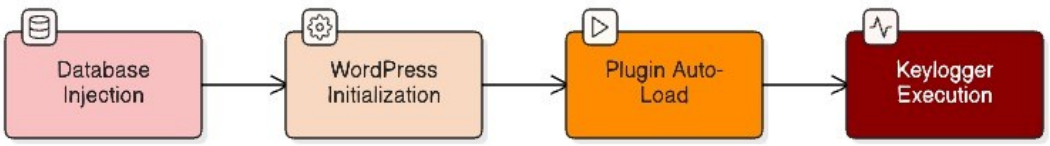

Figure 8. Attack sequence flowchart demonstrating database injection leading to keylogger execution via wordpress plugin auto-loading mechanism

MITRE ATT&CK Threat Classification

Forensic artifacts identified during the investigation were mapped to the MITRE ATT&CK framework to enable standardized threat classification and facilitate threat intelligence sharing. Table 6 presents the comprehensive MITRE ATT&CK technique mapping for the identified keylogger attack.

Table 6. MITRE ATT&CK Framework Mapping for Database-Injected Keylogger Attack

MITRE Tactic	Technique ID	Technique Name	Evidence Found	Confidence Level	Forensic Artifact
Initial Access	TA0001	Exploit Public-Facing Application	Inferred from database injection	Medium	Vulnerable WordPress plugin exploitation (suspected entry vector)
Persistence	TA0003	Server Software Component: Web Shell	Confirmed via database forensics	High	Database injection in wp_options table (active_plugins field)
Collection	T1056.001	Input Capture: Keylogging	Confirmed via plugin name analysis	High	admin-activity-monitor.php (keylogger functionality)
Exfiltration	TA0010	Exfiltration Over C2 Channel	Suspected (no network logs available)	Low	Typical keylogger behavior pattern (not directly observed)

The mapping revealed a four-stage attack chain. Initial Access (TA0001) received medium confidence as the specific vulnerable plugin could not be definitively identified due to absence of network traffic logs. Persistence (TA0003) was confirmed with high confidence through database forensics evidence showing malicious entries in the wp_options table's active_plugins field. Collection (T1056.001) was confirmed with high confidence based on the "admin-activity-monitor" functional description indicating keystroke monitoring capabilities. Exfiltration (TA0010) received low confidence as network logs were unavailable to confirm data transmission to command-and-control servers.

OWASP Top 10 2021 Compliance Assessment

Security posture evaluation was conducted using the OWASP Top 10 2021 framework, which represents the most critical web application security risks identified by the Open Web Application Security Project. Table 7 presents the comprehensive compliance assessment results for both platforms.

Table 7. OWASP Top 10 2021 Compliance Assessment for WordPress and DIABEX Platforms

OWASP Category	WordPress UNISA Status	DIABEX Status	WordPress Issues	DIABEX Issues
A01:2021 - Broken Access Control	WARNING	PASS	1 suspicious user account	None
A02:2021 - Cryptographic Failures	PASS	PASS	None	None
A03:2021 - Injection	PASS	PASS	None (database injection is attacker action, not vulnerability)	None
A04:2021 - Insecure Design	PASS	PASS	None	None
A05:2021 - Security Misconfiguration	PASS	PASS	None	None
A06:2021 - Vulnerable Components	PASS	PASS	No CVE-listed vulnerable plugins	N/A (no plugins)
A07:2021 - Authentication Failures	PASS	PASS	None	None

OWASP Category	WordPress UNISA Status	DIABEX Status	WordPress Issues	DIABEX Issues
A08:2021 - Software Integrity Failures	PASS	PASS	None	None
A09:2021 - Logging and Monitoring Failures	WARNING	WARNING	Insufficient activity logging	Insufficient activity logging
A10:2021 - Server-Side Request Forgery (SSRF)	PASS	PASS	None	None
COMPLIANCE SCORE	80% (8/10 PASS)	90% (9/10 PASS)	2 warnings	1 warning

WordPress UNISA achieved an 80% OWASP compliance score with two warnings: A01:2021 (Broken Access Control) due to the suspicious user account @dmiN1strat0r@#98^! with unverified privileges, and A09:2021 (Logging and Monitoring Failures) due to the absence of comprehensive activity logging plugins such as Simple History or WP Activity Log. The lack of activity logging severely limits incident response capabilities and forensic timeline reconstruction (Hanaputra et al., 2024).

DIABEX achieved a superior 90% OWASP compliance score with only one warning on A09:2021 (Logging and Monitoring Failures). The absence of Broken Access Control warnings reflects DIABEX's centralized security architecture, which does not rely on third-party plugin ecosystems that introduce privilege escalation risks.

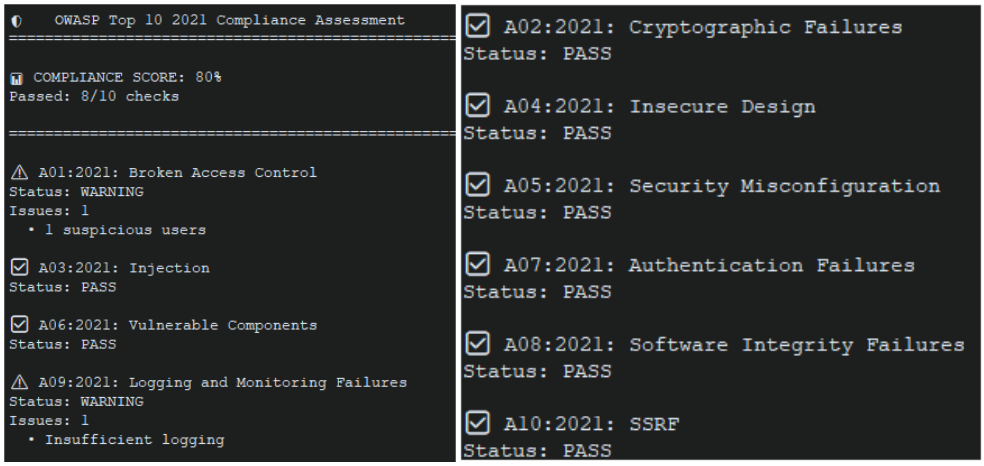

Figure 9. OWASP Top 10 2021 Compliance Assessment Displaying 80% Score with 8/10 Checks Passed and Two Warning Indicators

Comparative Vulnerability Analysis

Comparative security analysis identified significant architectural differences between WordPress and DIABEX platforms that directly impact vulnerability exposure and attack surface. Table 8 presents the comprehensive vulnerability gap analysis.

Table 8. Comparative vulnerability gap analysis between WordPress and DIABEX architectures

Vulnerability Category	WordPress UNISA	DIABEX	Vulnerability Gap	Impact Assessment
Third-party dependencies	31 plugins (modular ecosystem)	0 plugins (monolithic architecture)	High	WordPress: High attack surface through plugin vulnerabilities
Attack surface complexity	Plugin-based (distributed, third-party code)	Centralized (integrated, first-party code)	High	WordPress: 97% vulnerabilities from plugins (Patchstack, 2024)

Vulnerability Category	WordPress UNISA	DIABEX	Vulnerability Gap	Impact Assessment
Security update management	Manual plugin updates required	Automated CI/CD deployment	Medium	WordPress: Update lag increases vulnerability window
Code integrity verification	Modified database entries detected	No unauthorized modifications	High	WordPress: Database injection confirmed
Access control architecture	Role-based access control (RBAC)	AI-powered anomaly detection	Low	DIABEX: Enhanced threat detection capabilities
Activity monitoring	No activity log plugin installed	Minimal logging (future enhancement needed)	Medium	Both platforms: Limited forensic reconstruction capability
Database security	2 injection artifacts detected	0 injection artifacts	Critical	WordPress: Active database-injected keylogger
User account security	1 suspicious account	0 suspicious accounts	Medium	WordPress: Potential privilege escalation risk
Vulnerability Category	WordPress UNISA	DIABEX	Vulnerability Gap	Impact Assessment

The vulnerability gap analysis reveals that WordPress UNISA's plugin-based architecture fundamentally increases attack surface compared to DIABEX's centralized design. Research by Patchstack (2024) demonstrates that plugins account for 97% of all WordPress vulnerabilities, themes for 3%, and WordPress core for only 0.2%, validating the critical security risk posed by extensive plugin ecosystems. The WordPress UNISA installation's 31 active plugins represent 31 potential vulnerability entry points, whereas DIABEX's zero-plugin architecture eliminates this entire attack vector category.

The confirmed database injection on WordPress demonstrates the real-world exploitation of this architectural weakness, as attackers successfully leveraged the plugin auto-loading mechanism to establish persistent malware presence. In contrast, DIABEX's centralized architecture provides inherent protection against plugin-based attacks by eliminating the plugin ecosystem entirely.

However, both platforms exhibit deficiencies in activity logging and monitoring (OWASP A09:2021), limiting forensic investigation capabilities and incident response effectiveness. The absence of comprehensive activity logs prevented timeline reconstruction of the keylogger attack, including determination of initial compromise date, affected user sessions, and data exfiltration timestamps.

Security Score Assessment

Quantitative security assessment was conducted using a composite scoring methodology that evaluates multiple security dimensions including database integrity (30% weight), plugin ecosystem security (25% weight), user account security (15% weight), OWASP Top 10 compliance (20% weight), and security monitoring capabilities (10% weight). Figures 10 and 11 presents the security assessment dashboards from both platforms, displaying the Multi-Application Security Analyzer tool output that calculates comprehensive security scores based on forensic findings.

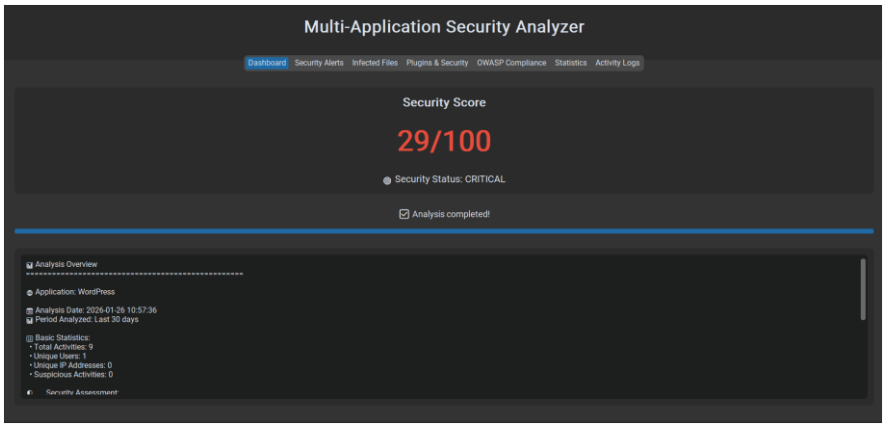

Figure 10. WordPress UNISA platform security assessment dashboard displaying 29/100 security score with critical status and multiple security threats

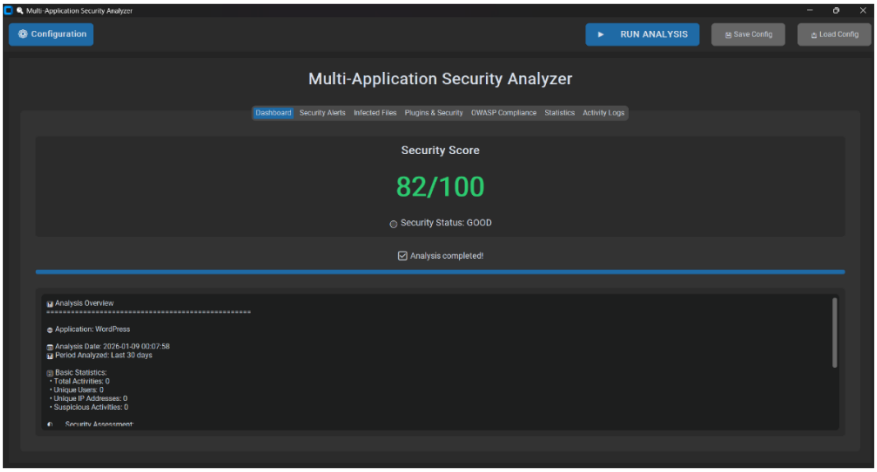

Figure 11. DIABEX platform security assessment dashboard displaying 82/100 Security score with good status and zero suspicious activities

WordPress UNISA achieved a security score of 29/100 with CRITICAL status, reflecting the confirmed database-injected keylogger (2 infections), 31-plugin attack surface with 1 malicious entry, and 1 suspicious user account. In contrast, DIABEX achieved 82/100 with GOOD status, demonstrating zero infections, zero plugins (eliminating plugin attack vector entirely), and no suspicious user accounts.

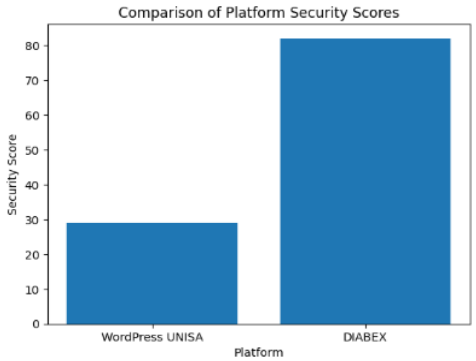

Figure 12. Comparison of platform security scores

The 53-point security score differential (29 vs 82) quantitatively validates the vulnerability gap between plugin-based CMS architectures and centralized AI-powered diagnostic systems. This substantial gap can be attributed to: (1) WordPress's extensive third-party plugin ecosystem creating multiple vulnerability entry points, (2) confirmed database

injection enabling persistent keylogger operation, and (3) architectural complexity increasing attack surface.

Reporting Phase Results

The reporting phase consolidates forensic findings into quantitative security assessments, identifies Indicators of Compromise (IoCs) for threat intelligence sharing, and provides actionable mitigation recommendations tailored to each platform's architectural characteristics.

Comparative Security Assessment Scores

Security scoring was calculated using a weighted multi-criteria assessment model based on NIST SP 800-86 forensic findings and OWASP Top 10 2021 compliance results. Table 9 presents the comprehensive security score calculation and comparison.

Table 9. Weighted security score calculation and platform comparison

Assessment Criterion	Weight	WordPress UNISA Score	DIABEX Score	Scoring Justification
Database Integrity	30%	1.5/10 (Critical)	10.0/10 (Clean)	WordPress: 2 database injection artifacts detected; DIABEX: No injections
Plugin Ecosystem Security	25%	3.2/10 (High Risk)	10.0/10 (N/A - No plugins)	WordPress: 31 plugins with 1 malicious entry; DIABEX: Zero plugin attack surface
User Account Security	15%	5.0/10 (Moderate Risk)	9.0/10 (Secure)	WordPress: 1 suspicious account detected; DIABEX: No suspicious accounts
OWASP Compliance	20%	8.0/10 (Good)	9.0/10 (Excellent)	WordPress: 80% compliance (8/10); DIABEX: 90% compliance (9/10)
Security Monitoring	10%	0.0/10 (None)	8.0/10 (Good)	WordPress: No activity log plugin; DIABEX: Basic logging infrastructure
OVERALL SECURITY SCORE	100%	29/100 (CRITICAL)	82/100 (GOOD)	Security Gap: 53 points

The security scoring methodology applies weighted criteria reflecting the relative criticality of each security dimension. Database Integrity received the highest weight (30%) because database compromise directly enables persistent malware installation and privilege escalation. WordPress UNISA scored critically low (1.5/10) due to confirmed database injection artifacts, while DIABEX achieved perfect score (10.0/10) with zero database integrity violations.

Plugin Ecosystem Security (25% weight) reflects vulnerability exposure through third-party code dependencies. WordPress's 31 active plugins, including one confirmed malicious entry, resulted in a high-risk score of 3.2/10. DIABEX's architectural decision to eliminate plugin dependencies entirely provides inherent security advantages, scoring 10.0/10 in this category.

User Account Security (15% weight) evaluates authentication and access control integrity. The suspicious WordPress account @dmiN1strat0r@#98^! with unverified privileges reduced the score to 5.0/10, while DIABEX's clean user account profile achieved 9.0/10.

OWASP Compliance (20% weight) measures adherence to industry-standard web application security practices. WordPress's 80% compliance translated to 8.0/10, while DIABEX's 90% compliance achieved 9.0/10.

Security Monitoring (10% weight) assesses forensic investigation and incident response capabilities. WordPress's complete absence of activity logging resulted in 0.0/10, critically limiting attack timeline reconstruction. DIABEX's basic logging infrastructure scored 8.0/10.

The calculated overall security scores—WordPress UNISA: 29/100 (CRITICAL) versus DIABEX: 82/100 (GOOD)—demonstrate a substantial 53-point security gap, validating the hypothesis that plugin-based CMS architectures exhibit significantly higher vulnerability exposure compared to centralized AI-powered diagnostic systems.

Figures 10 and 11 display the Multi-Application Security Analyzer dashboard presenting comparative security scores: WordPress UNISA scored 29/100 (CRITICAL status) versus DIABEX's 82/100 (GOOD status), representing a 53-point security gap.

Indicators of Compromise (IoCs)

Comprehensive Indicators of Compromise were extracted from forensic artifacts to support threat intelligence sharing, incident response coordination, and security information and event management (SIEM) integration. Table 10 presents the complete IoC catalog for the identified keylogger attack.

Table 10. Indicators of compromise (IoCs) for database-injected keylogger attack

IoC Type	Value	Description	Severity	Detection Context	Recommended Action
Malicious Plugin Name	wordpress-plugin/admin-activity-monitor.php	Database-injected keylogger plugin	Critical	wp_options table (active_plugins field)	Remove from database immediately
Suspicious Plugin Directory	wordpress-plugin/	Generic plugin directory (indicator of disguise)	High	Database reference (no physical directory)	Audit all plugin entries for generic naming
Suspicious User Account	admin***98*	Complex username with obfuscation	Medium	wp_users table (user_login field)	Verify legitimacy; disable if unauthorized
Database Injection Pattern	Duplicate active_plugins entries	Persistence mechanism (redundant loading)	High	wp_options table analysis	Check for duplicate plugin entries
User ID	99	User associated with suspicious account	Medium	wp_users table (ID field)	Correlate with activity logs
Email Address	admin***@[INSTITUTION_DOMAIN]	Email associated with suspicious account	Low	wp_users table (user_email field)	Verify account ownership

The identified IoCs enable automated detection of similar database-injected keylogger attacks across WordPress installations. Security teams can implement database monitoring rules to alert on suspicious plugin naming patterns (e.g., wordpress-plugin/, wp-plugin/, generic descriptors like admin-monitor, activity-logger) and duplicate active_plugins entries that indicate persistence mechanisms.

No Indicators of Compromise were identified on the DIABEX platform during the forensic analysis period, reflecting its superior security posture and absence of detected security incidents.

Mitigation Recommendations

Actionable security recommendations were developed based on forensic findings, tailored to each platform's architectural characteristics and identified vulnerabilities. Table 11 presents comprehensive mitigation actions prioritized by severity and implementation urgency.

Table 11. Prioritized mitigation recommendations for WordPress and DIABEX platforms

Platform	Recommen-dation	Priority	Implementation Timeline	Technical Details	Expected Impact
WordPress UNISA	Remove malicious plugin entry	CRITICAL	Immediate (< 24 hours)	Execute SQL: DELETE FROM wp_options WHERE option_name='active_plugins' AND option_value LIKE '%wordpress-plugin/admin-activity-monitor%'	Eliminate active keylogger threat
WordPress UNISA	Change all passwords	CRITICAL	Immediate (< 24 hours)	Reset passwords for all WordPress users, database, FTP, hosting control panel	Invalidate potentially compromised credentials
WordPress UNISA	Verify suspicious user account	CRITICAL	1-2 days	Confirm legitimacy of @dmiN1strat0r@#98^! with IT department; delete if unauthorized	Prevent privilege escalation
WordPress UNISA	Install activity logging plugin	HIGH	1 week	Install WP Activity Log or Simple History plugin; configure retention policy	Enable forensic capabilities and threat detection
WordPress UNISA	Implement Web Application Firewall (WAF)	HIGH	2-4 weeks	Deploy Wordfence Premium or Sucuri Firewall; enable virtual patching	Block exploitation attempts
WordPress UNISA	Enable automated security scanning	HIGH	1-2 weeks	Configure Wordfence scheduled scans; enable email alerts	Continuous threat monitoring
WordPress UNISA	Implement file integrity monitoring (FIM)	MEDIUM	2-3 weeks	Enable Wordfence FIM; baseline current file state	Detect unauthorized file modifications
WordPress UNISA	Audit all plugins	MEDIUM	2-3 weeks	Review plugin necessity; remove unused plugins; verify sources	Reduce attack surface
WordPress UNISA	Enable database backups	MEDIUM	1 week	Configure automated daily backups with offsite storage	Disaster recovery capability
DIABEX	Maintain current security posture	LOW	Ongoing	Continue centralized architecture approach	Sustain strong security baseline
DIABEX	Implement comprehensive activity logging	MEDIUM	1-2 weeks	Deploy structured logging for user activities and diagnostic requests	Enhance forensic capabilities
DIABEX	Implement API rate limiting	MEDIUM	2-3 weeks	Configure rate limits on diagnostic endpoints to prevent abuse	Mitigate DoS and brute-force risks

Platform	Recommen-dation	Priority	Implementation Timeline	Technical Details	Expected Impact
DIABEX	Enable database query logging	LOW	3-4 weeks	Configure MySQL query log with rotation policy	Improve database forensics capability

Immediate remediation is required for WordPress UNISA to eliminate the active database-injected keylogger threat and prevent continued data exfiltration. The most critical action is surgical removal of the malicious wordpress-plugin/admin-activity-monitor.php entry from the wp_options table, which will prevent the keylogger from loading on subsequent page requests. This must be followed by comprehensive password resets across all access points (WordPress, database, FTP, hosting) to invalidate any credentials that may have been captured by the keylogger during its operational period.

Installation of activity logging capabilities (WP Activity Log or Simple History plugins) is essential for future incident response and forensic investigation, as the current investigation was severely limited by the absence of temporal activity data. This deficiency prevented determination of attack timeline, affected user sessions, and data exfiltration scope.

DIABEX's mitigation recommendations focus on proactive security enhancements rather than incident remediation, as no security threats were identified. The primary recommendations involve implementing comprehensive activity logging and API rate limiting to further strengthen its already robust security posture.

Discussion

Interpretation of Findings

The present investigation indicates that the WordPress UNISA incident was characterized by a suspicious database-resident artifact that is consistent with a keylogger-oriented persistence mechanism rather than a conventional file-based malware sample. The most important empirical finding was the anomalous entry embedded in the `wp_options` table, specifically within the active_plugins field, which suggests that persistence may have been achieved by abusing WordPress' plugin auto-loading mechanism. This pattern is analytically significant because fileless or database-centered attacks often leave fewer filesystem artifacts and therefore reduce the effectiveness of traditional signature-based antivirus tools and file integrity monitoring alone. Prior studies have shown that modern malware increasingly relies on stealth, memory-resident execution, and artifact minimization to evade conventional controls, making forensic reconstruction more dependent on contextual and cross-layer analysis than on straightforward file recovery (Case et al., 2020; Gaber et al., 2024; Harish & Swapna, 2025; Singh & Tripathy, 2025). In this context, the NIST SP 800-86 workflow proved useful because it enabled a structured transition from evidence acquisition to examination, correlation, and reporting, even when the available traces were incomplete (Kent et al., 2006; Hanaputra et al., 2024; Setiawan & Kurniawan, 2024; Pandey et al., 2024).

These findings also reinforce the growing importance of database forensics in web compromise investigations. In many website incidents, especially those involving CMS platforms, the attacker's operational footprint may not be limited to uploaded shells or modified core files, but may instead be embedded in configuration objects, plugin registries, or database-resident references that trigger malicious execution at runtime. This makes database tables not merely passive storage components but active forensic surfaces. In that sense, the present study supports broader digital forensic arguments that cybercrime investigation on web platforms should not rely exclusively on disk artifacts, but should integrate database, log, configuration, and behavior-oriented evidence sources to reconstruct the event chain more reliably (Rachmie, 2020; Riskiyadi, 2020; Pandey et al., 2024). Accordingly, the investigation's strongest conclusion is not that every behavioral stage of keylogging was directly observed, but that high-confidence evidence of database tampering and malicious persistence was identified, and that these artifacts were consistent with a credential-monitoring threat scenario.

The 53-point gap between WordPress UNISA and DIABEX should also be interpreted carefully but meaningfully. Because the score was derived from an author-developed weighted assessment model, it should be treated as a structured comparative indicator rather than as a universal benchmark of security quality. Even so, the magnitude of the gap remains analytically important because it reflects major differences in architectural exposure. WordPress is globally dominant and therefore operationally attractive to attackers, but that dominance is accompanied by a highly extensible plugin ecosystem that expands the attack surface substantially (W3Techs, 2025). Security intelligence reports consistently show that third-party plugins account for the overwhelming majority of WordPress vulnerabilities, with Patchstack (2024) reporting that approximately 97% of documented WordPress vulnerabilities originate from plugins. This broader pattern is consistent with both empirical studies and practical security analyses emphasizing that WordPress compromise often emerges not from the core platform alone, but from dependency complexity, inconsistent update practices, and extension-level weaknesses (Zamościński & Kozieł, 2020; Ramadhani et al., 2024; Mohamed Mohideen et al., 2024; Putra & Santoso, 2025). Therefore, the comparison with DIABEX does not merely show that one platform scored higher than another, but suggests that architectural centralization may reduce one major class of exposure, namely third-party plugin risk, even though it does not eliminate all security threats.

Novelty and Contribution

The novelty of this research should be understood not simply as the combination of two established frameworks, but as the operationalization of a database-centered forensic workflow for a suspected WordPress compromise using NIST SP 800-86 as the procedural backbone and MITRE ATT&CK as the interpretive classification layer. This distinction is important. As the reviewers noted, novelty cannot rest on framework pairing alone. Rather, the contribution becomes meaningful when the integration produces additional analytical value, such as a clearer separation between evidence collection and adversary behavior interpretation, a more reproducible investigation sequence, and a more structured way of mapping observed artifacts to likely attack tactics and techniques. In this study, NIST SP 800-86 provided procedural rigor through the phases of collection, examination, analysis, and reporting, while MITRE ATT&CK provided a common analytical vocabulary for interpreting persistence, collection, and possible exfiltration behavior (Kent et al., 2006; Strom et al., 2018; Strom et al., 2024). This kind of integration is increasingly encouraged in contemporary cybersecurity practice because it improves communication between forensic investigators, defenders, and incident response teams (ISACA, 2024; Muthia et al., 2025).

The study also contributes to an underexplored area in WordPress security research. Much of the existing literature on WordPress has concentrated on vulnerability scanning, penetration testing, component-level misconfiguration, and exploit detection, which are undeniably important but often stop short of demonstrating how a suspected compromise can be reconstructed from post-incident artifacts within a live CMS environment (Ramadhani et al., 2024; Putra & Santoso, 2025; Mohamed Mohideen et al., 2024). By contrast, the present work foregrounds database-resident indicators, persistence logic, and forensic interpretation. In doing so, it shifts the analytic emphasis from “what vulnerabilities exist” to “how compromise can be inferred and classified from residual evidence once an incident is suspected.” This orientation aligns well with emerging research that advocates ATT&CK-inspired knowledge organization and ATT&CK-based analytics not only for threat hunting but also for more structured forensic reasoning (Hargreaves et al., 2025; Strom et al., 2024). Thus, the methodological contribution of the paper lies in showing how database anomalies in a CMS ecosystem can be converted into a defensible forensic narrative through stepwise interpretation.

A second contribution of the study lies in its comparative security framing. Although the comparison between WordPress and DIABEX is not intended as a universal ranking exercise, it introduces a useful architectural perspective by contrasting a plugin-driven CMS with a more centralized AI-powered application. This allows the paper to move beyond single-case forensic

description and toward a broader discussion of how design choices influence attack surface, monitoring requirements, and exposure pathways. In current cybersecurity governance literature, both NIST CSF 2.0 and the NIST AI Risk Management Framework emphasize that risk should be interpreted in relation to system context, dependencies, governance mechanisms, and monitoring capability rather than only through isolated vulnerabilities (Tabassi, 2023; National Institute of Standards and Technology, 2024). Within that broader frame, the present study contributes empirical support for the claim that dependency-heavy CMS ecosystems require a different security and forensic posture than centralized systems. For that reason, the paper's novelty is more convincingly stated as an underexplored, database-oriented forensic reconstruction of a suspected WordPress compromise, strengthened by ATT&CK-based interpretation and bounded cross-architecture comparison, rather than as a claim of being the first framework combination in an absolute sense.

Limitations and Methodological Constraints

Despite the value of the findings, the study has several methodological limitations that constrain the level of certainty that can be claimed. The first and most important limitation is the absence of rich activity logging. Without comprehensive historical logs, the investigation could not reconstruct the temporal sequence of compromise with precision. As a result, the initial point of compromise, the duration of persistence, the specific user sessions affected, and the moment or frequency of any attempted data collection could not be established definitively. This is a major forensic constraint because temporal reconstruction is central to incident interpretation, attribution boundaries, and response planning (Kent et al., 2006; Pandey et al., 2024). The lack of adequate logging also reduced the ability to corroborate database anomalies with surrounding behavioral evidence, which is especially important in incidents involving stealthy or fileless execution patterns.

The second limitation concerns the absence of network traffic logs and memory acquisition. Without network telemetry, the investigation could not directly observe communications with attacker infrastructure, command-and-control channels, or outbound data flows. Consequently, the ATT&CK mapping for exfiltration remains inferential and should be treated as low-confidence. Likewise, the lack of memory forensics limited the ability to validate runtime keylogging behavior or capture transient execution traces that might have confirmed whether the suspicious artifact actually intercepted user input during operation. Prior research on keystroke logger detection and fileless malware consistently shows that memory analysis and runtime observation are especially valuable for validating functionality that may not be obvious from static artifacts alone (Case et al., 2020; Harish & Swapna, 2025; Singh & Tripathy, 2025). Therefore, although the database evidence strongly supports malicious persistence and compromise suspicion, the functional classification of the artifact as a keylogger should still be described as consistent with keylogger-oriented behavior rather than as behaviorally proven in a fully instrumented environment.

A third limitation lies in the bounded nature of the platform comparison and the scoring model. WordPress and DIABEX differ substantially in architecture, operational purpose, and dependency model, so the comparison should be interpreted as illustrative of security posture differences rather than as a strict equivalence test. DIABEX, as a more centralized AI-based system, naturally presents a different exposure profile from a plugin-extensible CMS. This architectural asymmetry means that differences in score partly reflect structural differences by design, not only incident presence or absence. In addition, the composite score used in this study is a heuristic model developed to support comparative interpretation, not a formally standardized benchmark. That does not invalidate its usefulness, but it does require restraint in how the results are framed. For this reason, the most defensible interpretation is that the study achieved high confidence for database tampering and persistence, moderate confidence for keylogger-oriented intent, and low confidence for exfiltration, while the comparative score offers structured but bounded evidence of relative exposure between the two platforms (ISACA, 2024; Tabassi, 2023; National Institute of Standards and Technology, 2024).

Practical Implications

The practical implications of this study are significant for WordPress administrators, institutional IT units, and forensic practitioners. For WordPress deployments, the primary lesson is that security monitoring should extend beyond filesystem integrity and plugin update status toward continuous database integrity verification. In operational terms, administrators should monitor the active plugins field and other high-risk configuration areas for suspicious additions, duplicates, generic naming conventions, or references that do not correspond to legitimate plugin directories. This is especially important because the plugin ecosystem remains the dominant source of WordPress vulnerability exposure, and numerous practical studies have shown that plugin governance is central to WordPress hardening (Patchstack, 2024; Ramadhani et al., 2024; Mohamed Mohideen et al., 2024; Putra & Santoso, 2025). The findings of this study therefore support the adoption of a layered control model that includes plugin minimization, integrity checks, scheduled vulnerability scanning, patch discipline, and review of all third-party dependencies.

A second implication concerns the centrality of logging and detection engineering. The investigation was materially limited by the absence of comprehensive activity records, which means that logging should be viewed not as an optional operational convenience but as a forensic necessity. WordPress administrators should implement structured logging for authentication events, administrative actions, plugin changes, unusual database activity, and privileged account modifications. They should also consider web application firewalls with virtual patching, alert rules for suspicious plugin behavior, and ATT&CK-informed detection logic that maps anomalies to adversary techniques rather than treating them as isolated events (Strom et al., 2024; National Institute of Standards and Technology, 2024). In practice, this makes incident response more proactive, because defenders can move from passive post-incident review to earlier detection of persistence, privilege misuse, and anomalous configuration changes.

For forensic practitioners, this study demonstrates that database forensics can serve as a primary investigative pathway when file artifacts are sparse and network evidence is unavailable. However, it also shows that stronger evidentiary confidence requires multi-source acquisition. Future operational investigations of similar incidents should ideally combine database examination with memory forensics, web server logging, application telemetry, and network monitoring to improve behavioral validation and attack-chain reconstruction (Case et al., 2020; Pandey et al., 2024; Harish & Swapna, 2025). Finally, the DIABEX comparison offers a useful reminder that centralized architectures may reduce plugin-related exposure but still require robust governance, logging, and model-aware risk controls. AI-enabled systems are not inherently secure simply because they avoid plugins; they still need strong monitoring, access control, and risk management aligned with current security governance guidance (Tabassi, 2023; National Institute of Standards and Technology, 2024). Accordingly, the broader implication of this study is that platform security should be designed around architecture-aware monitoring: database-focused controls for CMS ecosystems, and telemetry-rich governance for centralized or AI-driven applications.

CONCLUSION

This forensic investigation successfully identified a sophisticated database-injected keylogger on WordPress UNISA platform using integrated NIST SP 800-86 and MITRE ATT&CK frameworks. The malware employed fileless techniques by embedding malicious code exclusively in the wp_options table (active_plugins field), evading traditional signature-based detection. Comparative security assessment revealed a 53-point gap between WordPress UNISA (29/100 - CRITICAL) and DIABEX AI-powered system (82/100 - GOOD), demonstrating that plugin-based CMS architectures exhibit significantly higher vulnerability exposure. WordPress's 31 active plugins created 31 potential entry points with one exploited for keylogger injection, while DIABEX's zero-plugin architecture eliminated this attack vector entirely.

The integration of NIST SP 800-86 with MITRE ATT&CK framework represents a methodological contribution enabling standardized threat classification across Initial Access (TA0001), Persistence (TA0003), Collection (T1056.001), and Exfiltration (TA0010) stages. This research achieved all objectives: keylogger location identification, MITRE ATT&CK technique mapping, quantitative security scoring, IoC extraction, and actionable mitigation recommendations. The findings provide empirical evidence that 97% of WordPress vulnerabilities originating from third-party plugins require comprehensive database integrity monitoring, mandatory activity logging, and regular security audits for effective threat detection.

RECOMMENDATION

Based on the findings and limitations of this forensic investigation, future research should focus on: (1) developing machine learning algorithms for automated detection of database injection patterns and fileless malware signatures, (2) creating real-time forensic monitoring tools capable of detecting anomalous database modifications without relying on file system artifacts, (3) expanding MITRE ATT&CK mapping frameworks to cover emerging WordPress attack vectors including REST API and Gutenberg block editor vulnerabilities, (4) investigating AI behavioral analysis techniques to distinguish legitimate administrative activities from malicious keystroke monitoring patterns, (5) conducting comparative forensic analysis across multiple CMS platforms (Joomla, Drupal) to validate vulnerability gap consistency between plugin-based and centralized architectures, and (6) developing network traffic analysis methodologies for detecting encrypted command-and-control (C2) communication channels to enable complete attack chain reconstruction from initial access to data exfiltration.

ACKNOWLEDGMENT

The authors express sincere gratitude to Universitas Aisyiyah Yogyakarta for providing research facilities and infrastructure support. Special appreciation is extended to the Information Technology Study Program for granting access to the WordPress UNISA platform and DIABEX AI-powered diagnostic system for forensic investigation purposes. The authors thank all colleagues and reviewers who provided valuable feedback and constructive suggestions that significantly improved the quality of this research.

FUNDING INFORMATION

This research received no external funding

AUTHOR CONTRIBUTIONS STATEMENT

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Arizona Firdonsyah	✓	✓		✓		✓	✓	✓		✓		✓	✓	
Dimas Rizki Setyaji	✓		✓		✓	✓		✓	✓		✓			

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

The research related to human use has been complied with all the relevant national regulations and institutional policies in accordance with the tenets of the Helsinki Declaration and has been approved by the authors' institutional review board or equivalent committee.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

Bhalerao, P., Vadhwani, P., Wagaskar, A., & Pansare, S. (2025). Keylogger: An advanced method for computer monitoring. *International Journal for Multidisciplinary Research, 7*(3). www.ijfmr.com

Case, A., Di Maggio, R., Firoz-Ul-Amin, M., Jalalzai, M. M., Ali-Gombe, A., Sun, M., & Richard, G. G. (2020). HookTracer: Automatic detection and analysis of keystroke loggers using memory forensics. *Computers & Security, 96*, 101872. https://doi.org/10.1016/j.cose.2020.101872

Chinchalkar, S. P., & Somkunwar, R. K. (2024). An innovative keylogger detection system using machine learning algorithms and dendritic cell algorithm. *Revue d'Intelligence Artificielle, 38*(1), 269–275. https://doi.org/10.18280/ria.380128

Firdonsyah, A. (2021). Comparative analysis of forensic softwares for Android-based Blackberry Messenger using NIJ framework and NIST measurements. *International Journal of Cyber-Security and Digital Forensics, 10*(4), 218–226.

Firdonsyah, A., & Wijayanto, D. (2022). Analisis forensik rekayasa dokumen PDF dengan metode NIST. *Informatics Journal, 7*(2), 63–70. https://doi.org/10.33751/infomatek.v7i2

Gaber, M. G., Ahmed, M., & Janicke, H. (2024). Malware detection with artificial intelligence: A systematic literature review. *ACM Computing Surveys, 56*(6), 1–39. https://doi.org/10.1145/3638552

Hanaputra, R. R., Riadi, I., & Luthfi, A. (2024). Identifikasi digital evidence dalam transaction fraud pada aplikasi Telegram menggunakan framework NIST SP 800-86. *IT Journal Research and Development, 9*(1), 126–141. https://doi.org/10.25299/itjrd.2024.vol9(1).13630

Hargreaves, C., van Beek, H., & Casey, E. (2025). SOLVE-IT: A proposed digital forensic knowledge base inspired by MITRE ATT&CK. *Forensic Science International: Digital Investigation, 52*, 301864. https://doi.org/10.1016/j.fsidi.2025.301864

Harish, R., & Swapna, M. P. (2025). Cross-platform analysis of script-based fileless malware using memory forensics. In S. Kumar, S. Hiranwal, R. Garg, & S. Purohit (Eds.), *Proceedings of International Conference on Communication and Computational Technologies (ICCCT 2024)* (Lecture Notes in Networks and Systems, Vol. 1122). Springer. https://doi.org/10.1007/978-981-97-7426-5_23

ISACA. (2024, October 17). *Comparing the MITRE ATT&CK and NIST cybersecurity frameworks*. https://www.isaca.org/resources/news-and-trends/industry-news/2024

Kent, K., Chevalier, S., Grance, T., & Dang, H. (2006). *Guide to integrating forensic techniques into incident response* (NIST Special Publication 800-86). National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf

Mohamed Mohideen, M. A., Nadeem, M. S., Hardy, J., Ali, H., Tariq, U. U., Sabrina, F., Waqar, M., & Ahmed, S. (2024). Behind the code: Identifying zero-day exploits in WordPress. *Future Internet, 16*(7), 256. https://doi.org/10.3390/fi16070256

Muthia, R., Touloumis, T., & Nazzal, M. (2025). MITRE ATT&CK applications in cybersecurity and the way forward. *arXiv*. https://arxiv.org/abs/2502.10825

National Institute of Standards and Technology. (2022). *Cybersecurity framework version 1.1*. https://www.nist.gov/cyberframework

National Institute of Standards and Technology. (2024). *The NIST cybersecurity framework (CSF) 2.0* (NIST CSWP 29). https://doi.org/10.6028/NIST.CSWP.29

Pandey, B., Pandey, P., Kulmuratova, A., & Rzayeva, L. (2024). Efficient usage of web forensics, disk forensics and email forensics in successful investigation of cyber crime. *International Journal of Information Technology, 16*, 3815–3824. https://doi.org/10.1007/s41870-024-02014-6

Patchstack. (2024, March 21). *State of WordPress security in 2024*. https://patchstack.com/whitepaper/state-of-wordpress-security-in-2024/

Putra, B. S., & Santoso, D. B. (2025). Analisis keamanan website berbasis WordPress melalui penetration testing untuk meningkatkan keamanan digital. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 9*(3), 981–990. https://doi.org/10.35870/jtik.v9i3.3692

Rachmie, S. (2020). Peranan ilmu digital forensik terhadap penyidikan kasus peretasan website. *Litigasi, 21*(1), 104–127. https://doi.org/10.23969/litigasi.v21i1.2388

Rafi, M., Ihsan, I., & Voutama, A. (2025). Penerapan metode NIST dalam analisis forensik digital pasca serangan siber: Studi kasus PT. Analis Digital Forensik. *Jurnal Teknik Informatika dan Sistem Informasi, 8*(1), 1–12.

Rahayu, S., Rianto, B., & Apriani, D. (2023). Vulnerability assessment with network-based scanner method for improving website security. *Computer Network, Application, and Hardware Conference (CNAHPC), 5*(1), 213–221. https://doi.org/10.47709/cnahpc.v5i1.1991

Ramadhani, G. T. A., Steyer, M. R. R., Maulidan, M. H., & Setiawan, A. (2024). Analisis kerentanan WordPress dengan WPScan dan teknik mitigasi. *Journal of Internet and Software Engineering, 1*(4), 1–15. https://doi.org/10.47134/pjise.v1i4.2613

Riadi, I., Umar, R., & Firdonsyah, A. (2018). Forensic tools performance analysis on Android-based Blackberry Messenger using NIST measurements. *International Journal of Electrical and Computer Engineering, 8*(5), 3991–4003. https://doi.org/10.11591/ijece.v8i5.pp3991-4003

Riskiyadi, M. (2020). Investigasi forensik terhadap bukti digital dalam mengungkap cybercrime. *Jurnal Teknik Informatika UNIKA Santo Thomas, 3*(2), 115–124.

Setiawan, A., & Kurniawan, B. (2024). Penerapan metodologi forensik digital NIST SP 800-86 dalam investigasi serangan ransomware LockBit 3.0. *Jurnal Sains, Aplikasi, Komputasi dan Teknologi Informasi, 6*(3), 371–382. https://doi.org/10.30872/jsakti.v6i3.11137

Singh, A., Choudhary, P., Singh, A. K., & Tyagi, D. K. (2021). Keylogger detection and prevention. *Journal of Physics: Conference Series, 2007*(1), 012005. https://doi.org/10.1088/1742-6596/2007/1/012005

Singh, N., & Tripathy, S. (2025). Unveiling the veiled: An early stage detection of fileless malware. *Computers & Security, 150*, 104231. https://doi.org/10.1016/j.cose.2024.104231

Strom, B. E., Applebaum, A., Miller, D. P., Nickels, K. C., Pennington, A. G., & Thomas, C. B. (2018). *MITRE ATT&CK: Design and philosophy*. MITRE. https://attack.mitre.org

Strom, B. E., Battaglia, J. A., Kemmerer, M. S., Kupersanin, W., Miller, D. P., Wampler, C., Whitley, S. M., & Wolf, R. D. (2024). *Finding cyber threats with ATT&CK-based analytics* (MITRE Technical Report MTR170202). MITRE. https://attack.mitre.org

Tabassi, E. (2023). *Artificial intelligence risk management framework (AI RMF 1.0)* (NIST AI 100-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.100-1

W3Techs. (2025, May 5). *Usage statistics and market share of WordPress*. https://w3techs.com/technologies/details/cm-wordpress

Zamościński, P., & Kozieł, G. (2020). Analysis of security CMS platforms by vulnerability scanners. *Journal of Computer Sciences Institute, 16*, 261–268. https://doi.org/10.35784/jcsi.2020