RESEARCH ARTICLE OPEN ACCESS

ANALYSIS OF CREDENTIAL STUFFING & PASSWORD SPRAYING ATTACK MODELS AND A DEFENSE-IN-DEPTH STRATEGY FOR HIGHER EDUCATION MANAGEMENT SYSTEMS

Nguyen Thi Hong Mai[1], Nguyen Manh Hung[1], Bui Thi Thuy Quynh[1]
[1]*Faculty of Information Tehcnlogy, Hai Phong University. Haiphong, Vietnam*
Emails: mainth@dhhp.edu.vn; hungnm@dhhp.edu.vn; quynhbtt@dhhp.edu.vn

Received: Revised: Accepted: Published:

Abstract - *In the digital transformation of higher education, academic management portals and Learning Management Systems (LMS) have become primary targets for automated cyberattacks. Although server-side password storage hashing algorithms such as Bcrypt or Scrypt have proven effective against database offline cracking, they remain largely ineffective against password enumeration attacks executed through legitimate authentication flows, specifically Credential Stuffing and Password Spraying. This paper presents an in-depth analysis of the theoretical models of Credential Stuffing and Password Spraying, highlighting system vulnerabilities and specific security risks inherent to the higher education environment. On this basis, the study proposes a 4-layer Defense-in-Depth Architecture consisting of independent yet complementary control layers, integrating security standards from OWASP and NIST. The research findings contribute a solid theoretical foundation to assist educational institutions in optimizing their information security infrastructure without compromising the digital user experience.*

Keywords - *Credential Stuffing, Password Spraying, Defense-in-Depth, Academic Management Systems, Higher Education Information Security, NIST, OWASP*

1. Introduction

The rapid explosion of digital transformation in higher education institutions has transformed Academic Management Systems (AMS) and student portals into centralized repositories for sensitive data, including Personal Identifiable Information (PII), academic records, financial data, and intellectual property. However, alongside this convenience comes an increasing risk of information insecurity. One of the most critical vulnerabilities in university digital infrastructure lies within the user authentication phase.

In previous studies, password protection primarily focused on server-side storage controls, typically employing adaptive hashing algorithms such as Bcrypt, Scrypt, or PBKDF2 to mitigate brute-force attacks in the event of a database breach. Nevertheless, operational reality indicates that modern attackers do not necessarily need to compromise the database to harvest credentials. Instead, they exploit valid authentication workflows (Authentication APIs/Forms) using large-scale automated attack techniques, such as **Credential Stuffing** (utilizing leaked credential lists from other breaches) and **Password Spraying** (testing highly common passwords across a wide array of accounts).

In a university setting, these two attack vectors are exceptionally dangerous due to user behavioral patterns: students frequently reuse passwords across social networks and entertainment services and tend to set predictable credentials. Therefore, designing a server-side Defense-in-Depth architecture that proactively detects and mitigates automated attacks directly at the authentication pipeline is an urgent and strategic requirement.

2. Theoretical Models and the Nature of Automated Attacks

2.1. Credential Stuffing Attacks

Credential Stuffing attacks exploit password reuse behavior across multiple services. Attackers acquire breach datasets containing compromised Username/Email–Password pairs (*Combo lists*) from major global data leaks, and subsequently utilize automated tools (Bots) to "stuff" millions of these credentials into the login portals of the target system.

Technical Execution Flow:

- **Data Harvest:** Acquiring credentials from leaked breach repositories (*Combo lists*).
- **Botnet & Proxy Rotation Deployment:** Utilizing botnets or residential proxy networks to distribute and throttle login requests, bypassing basic IP-based rate limiting.
- **Automated Execution:** Programmatically issuing automated HTTP POST requests containing credentials to the authentication API endpoints.

- **Response Analysis:** Parsing HTTP Status Codes or response payload lengths to identify successful login sessions.

2.2. Password Spraying Attacks

In contrast to traditional brute-force attacks (testing thousands of passwords against a single account, which rapidly triggers Account Lockout policies), Password Spraying adopts the opposite methodology: testing one or two extremely common passwords (e.g., 123456, Mover2024@, Haiphong2024) across thousands of distinct accounts.

Evasion Mechanism: Most academic management systems enforce account lockout policies after 3 to 5 consecutive failed login attempts within a short timeframe. Password Spraying bypasses this mechanism by introducing temporal delays between attempts on the same account (e.g., testing only one password every two hours per account). Because the failure frequency per individual account remains extremely low, standard security controls fail to trigger account lockouts, yet the aggregate attack achieves a high overall success rate across the institution.

2.3. Technical Comparison of Authentication Attack Vectors

Table 1. Technical comparison between traditional Brute-Force, Credential Stuffing, and Password Spraying attacks.

Evaluation Criteria	Traditional Brute-Force	Credential Stuffing	Password Spraying
Search Space	1 Account × N Passwords (exhaustive)	N Accounts × N Passwords (Combo list)	N Accounts × 1–2 Common Passwords
Data Source	Random Strings / Dictionaries	Breach "Combo lists" from external leaks	Weak Password Lists + Student ID Directory
Per-Account Request Frequency	Extremely High (Hundreds of req/sec)	Low to Moderate (Botnet dependent)	Extremely Low (1 attempt / several hours)
Primary Detection Indicator	Rapid, concentrated login failures on 1 user	System-wide surge in global failure rates	Scattered failures across many users from few IPs
Lockout Evasion Capability	Very Poor (Triggers lockout immediately)	Moderate	Very High (Difficult to detect via static rules)

3. Risk Assessment and System Vulnerabilities in Higher Education

Academic management systems in universities exhibit specific operational characteristics that create favorable environments for automated attacks:

- **Predictable Student ID / Username Structure:** Student IDs typically follow a rigid formatting convention (e.g., StudentID + Cohort + Major + Sequence). This enables attackers to easily harvest or generate directories of tens of thousands of valid usernames without discovery.
- **Default Password Habits among Freshmen:** Systems often assign default initial passwords based on birthdates (DDMMYYYY) or national identification numbers. Many students neglect to change these defaults, leaving a major vulnerability for Password Spraying campaigns at the beginning of academic terms.
- **Personal Password Reuse:** Students routinely reuse identical passwords across personal emails, social media, and university portals. When an external third-party application is breached, university accounts are immediately exposed to Credential Stuffing threats.
- **Absence of API-Side Security Controls:** Many portals focus security controls solely on Web HTML login forms while neglecting mobile app APIs and Web Service endpoints, allowing attackers to directly send high-throughput HTTP requests.

4. Proposed Defense-in-Depth Architecture

To effectively counter automated attacks without degrading the user experience (UX), this paper proposes a 4-layer Defense-in-Depth Architecture implemented at the university authentication gateway:

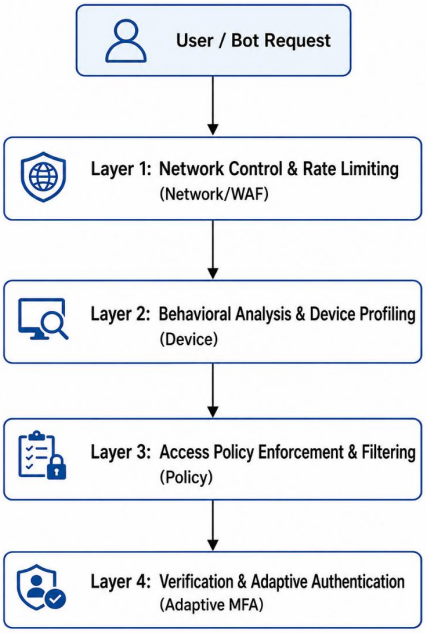

Fig. 1. The proposed 4-layer Defense-in-Depth Architecture protecting the authentication flow of Higher Education Management Systems.

4.1. Layer 1: Network Traffic Control and Smart Rate Limiting

The outermost layer serves as the first line of defense by monitoring and controlling abnormal traffic before authentication requests reach the application server. Its primary objective is to mitigate large-scale automated attacks, such as Credential Stuffing and Password Spraying, while reducing unnecessary resource consumption on the authentication infrastructure.

- Multi-dimensional Rate Limiting: Instead of relying solely on IP-based rate limiting, which can be easily bypassed through proxy rotation, the proposed framework applies rate limiting based on multiple attributes, including the combination of IP address and User-Agent, the /24 subnet, and the target username. This approach enables the system to detect distributed attack patterns in which requests originate from multiple IP addresses but target the same group of user accounts. Once predefined thresholds are exceeded, the system may introduce response delays, temporarily reject requests (e.g., HTTP 429 – Too Many Requests), or require CAPTCHA verification.
- IP Reputation Services: The framework integrates Web Application Firewall (WAF) with IP reputation services to assess the trustworthiness of incoming connections. Authentication requests originating from Tor Exit Nodes, public VPNs, or cloud-hosting IP ranges (e.g., AWS, DigitalOcean, and Microsoft Azure) are assigned a higher risk score and may be blocked or challenged with additional verification. By filtering high-risk traffic at the network perimeter, this layer significantly reduces the volume of malicious requests reaching the authentication server, thereby improving the overall effectiveness of the proposed defense-in-depth architecture.

4.2. Layer 2: Device Fingerprinting and Contextual Behavioral Analysis

This layer focuses on identifying abnormal client-side characteristics and behavioral patterns that cannot be detected through network traffic analysis alone. By combining Device Fingerprinting with Contextual Behavioral Analysis, the system can distinguish legitimate users from automated tools, even when attackers employ proxy rotation or distributed botnets to evade network-level defenses.

- **Device Fingerprinting:** The system collects multiple browser and device attributes, including browser hardware characteristics, installed fonts, Canvas rendering outputs, screen resolution, operating system, and browser configuration, to generate a unique Device Fingerprint for each client. Unlike IP addresses, which can be easily changed, device fingerprints remain relatively stable across login sessions, making them an effective indicator for detecting automated attacks. If a single fingerprint attempts to authenticate against hundreds of different user accounts within a short period, the system classifies the activity as a potential bot operation and automatically blocks or challenges subsequent authentication requests.
- **Geographic and Contextual Anomaly Detection:** In addition to device identification, the system continuously evaluates contextual information associated with each login attempt, including geographic location, login history, access time, and device consistency. Alerts are generated when abnormal behaviors are detected, such as impossible travel (e.g., a user successfully logs in from Hai Phong and then from another country five minutes

later), sudden changes in device characteristics, or repeated login attempts from previously unseen locations. These contextual indicators are incorporated into the overall risk assessment process, enabling the authentication system to respond dynamically to suspicious activities while minimizing disruption for legitimate users.

4.3. Layer 3: Password Policy Control and Breach Database API Integration

This layer focuses on strengthening password security by preventing the use of weak or previously compromised credentials. Its primary objective is to reduce the success rate of Credential Stuffing and Password Spraying attacks before authentication requests are processed.

- **Proactive Compromised Password Lookup:** Whenever a user creates or updates a password, the system automatically verifies its hashed value using the SHA-1 k-Anonymity protocol against publicly available breach databases, such as the Have I Been Pwned (HIBP) API. This privacy-preserving mechanism allows the system to determine whether the password has appeared in previous data breaches without transmitting the plaintext password. If a match is found, the password is rejected and the user is required to select a new one, thereby preventing the reuse of compromised credentials.
- **Anti-Password Spraying Password Policies:** The framework also enforces password policies designed to eliminate commonly targeted passwords used in Password Spraying attacks. Passwords containing the university name, current calendar year, student ID, date of birth, or common keyboard patterns (e.g., *123456*, *qwerty*, or *password*) are prohibited. In addition, the system requires users to create passwords with sufficient complexity and entropy in accordance with the recommendations of NIST SP 800-63B. These measures significantly reduce the effectiveness of dictionary-based attacks while improving the overall security of the authentication system.

4.4. Layer 4: Risk-Based Authentication

The fourth layer serves as the final decision-making component of the proposed defense-in-depth architecture. Instead of applying identical authentication policies to all users, the system dynamically adjusts security controls based on the estimated risk associated with each login attempt. This approach enhances security while minimizing unnecessary authentication challenges for legitimate users, thereby improving the overall user experience.

- **Real-Time Risk Assessment:** For every authentication request, the system calculates a Risk Score by aggregating security indicators collected from the previous layers, including IP reputation, rate-limiting events, device fingerprint consistency, geographic anomalies, login history, and password-related risks. The overall risk score can be expressed as:

$$R = w_1 \cdot IPRisk + w_2 \cdot DiviceRisk + w_3 \cdot BehaviorRisk + w_4 \cdot PasswordRisk$$

where:

- ***R*** denotes the overall authentication risk score for a login request.
- ***IPRisk*** represents the risk score associated with the source IP address, considering factors such as IP reputation, proxy usage, Tor exit nodes, and abnormal request frequency.
- ***DeviceRisk*** denotes the risk score derived from Device Fingerprinting, reflecting anomalies in browser characteristics, device consistency, or suspicious fingerprint reuse.
- **BehaviorRisk** represents the risk score calculated from behavioral and contextual analysis, including login history, geographic anomalies, login frequency, and unusual access patterns.
- ***w1, w2*** and ***w3*** are weighting coefficients assigned to each risk factor, where ***wi*** ≥ 0. These coefficients can be adjusted according to the security policy and operational requirements of the university's authentication system, with larger values indicating a greater contribution of the corresponding factor to the overall risk score.
- **PasswordRisk** represents the risk associated with the user's password, such as the use of weak, predictable, or previously compromised passwords identified through breach database verification.

- **Risk-Based Authentication Response:** Based on the calculated Risk Score, the authentication server dynamically selects an appropriate security response. Low-risk requests are processed through standard authentication without additional verification to ensure a seamless user experience. Medium-risk requests require additional verification, such as Invisible reCAPTCHA, One-Time Password (OTP), or email verification. High-risk requests are temporarily blocked, logged for security analysis, and may trigger alerts to system administrators or account owners. Furthermore, all high-risk authentication events are recorded in centralized security logs, enabling subsequent forensic analysis and supporting continuous improvement of detection rules. By applying security measures proportionally to the assessed risk, the proposed framework effectively mitigates automated attacks while avoiding unnecessary interruptions for legitimate users. This adaptive approach aligns with the recommendations of NIST SP 800-63B, which advocates risk-based authentication as an effective mechanism for balancing security, usability, and operational efficiency in modern identity management systems.

The weighting coefficients may be empirically determined based on historical authentication logs or optimized using machine learning techniques to improve the accuracy of risk assessment.

5. Discussion and Practical Implementation Implications

The proposed Defense-in-Depth Architecture yields significant operational benefits for university cybersecurity management:

- **Overcoming Database Hashing Limitations:** While Bcrypt and Scrypt only secure data-at-rest, this 4-layer model protects data-in-transit and prevents logic abuse at the application level.
- **User Experience (UX) Optimization:** By applying Adaptive Authentication, legitimate students logging in from trusted devices and campus networks experience zero friction (no CAPTCHAs or OTP prompts). Security challenges are step-up triggered only during anomalous events, achieving an optimal balance between security and UX.
- **High Feasibility and Interoperability:** Designed around open industry standards (OWASP Automated Threat Handbook, NIST SP 800-63B), the framework enables university IT departments to incrementally integrate these layers into legacy systems without altering underlying user database schemas.

6. Conclusion

Automated threats such as Credential Stuffing and Password Spraying represent major risks to student account security and system integrity in higher education environments. Sole reliance on server-side database password hashing is no longer sufficient. This paper has provided an in-depth analysis of these attack vectors and introduced a 4-layer Defense-in-Depth Architecture. The proposed solution enhances the security posture of higher education management platforms and offers a reference architecture for academic institutions pursuing secure, sustainable digital transformation.

References

[1] Le Dac Nhuong (2018), Data Security, Vietnam National University Press, Hanoi.
[2] Nguyen Van Tuan, Nguyen Hong Son (2022), Performance and Security Evaluation of Password Hashing Algorithms in E-Learning Systems, Journal of ICT.
[3] NIST (2020), Digital Identity Guidelines: Authentication and Lifecycle Management, NIST Special Publication 800-63B, National Institute of Standards and Technology.
[4] OWASP (2021), OWASP Automated Threat Handbook for Web Applications, Open Web Application Security Project.
[5] Shay, R., et al. (2016), Designing Password Policies for Strength and Usability, ACM Transactions on Information and System Security (TISSEC).