Mikailalsys Journal of
Advanced Engineering International

e-ISSN : 3046-9694
p-ISSN : 3046-8914

Index : Harvard, Boston, Sydney University, Dimensions, Scilit, Semantic, Google, etc

https://doi.org/10.58578/MJAEI.v3i3.12589

Security Assessment of PIN-Based Authentication Mechanisms in Web Applications: A Penetration Testing Case Study of a Healthcare Platform in Nigeria

Dadi Jonathan Abba[1], Habibu Aminu Sani[2], Chat Darius Chinyio[3], Darius Tienhua Chinyio[4]

[1]Karl Kumm University Vom, Plateau State, Nigeria; [2]Nuhu Bamali Polytechnic Zaria, Nigeria
[3,4]Nigerian Defence Academy Kaduna, Nigeria
jonathandadiabba@gmail.com

Submitted:	Revised:	Accepted:	Published:
Sep 25, 2026	Oct 24, 2026	Nov 4, 2026	Nov 9, 2026

Abstract

Healthcare web applications process sensitive operational and personal information, making secure authentication essential. This paper presents an authorized penetration-testing case study of the PIN-based administrator authentication mechanism in the development environment of MySmartMedic PHC Edition, drawing on evidence documented in an industrial training report. The assessment used controlled testing and evidence-based vulnerability reporting to examine the administrator login workflow, resistance to repeated PIN-verification attempts, and authentication token handling. The documented findings identified ineffective rate limiting and weak PIN complexity requirements. The report also recorded that successful verification using a weak PIN resulted in the application returning a valid bearer token. Together, weak PIN requirements and insufficient attempt controls could facilitate unauthorized access to an authenticated session if the reported conditions were reproducible in the assessed environment. These findings highlight the importance of layered authentication controls, server-side attempt throttling, secure token issuance and handling, and systematic security testing throughout the software development lifecycle. The case provides context-specific evidence to inform authentication

MJAEI Journal is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

security assessment in healthcare applications. Because the source material is an industrial training report rather than a controlled experiment, the findings do not establish vulnerability prevalence or statistical generalizability.

Keywords: PIN-Based Authentication; Penetration Testing; Rate Limiting; Bearer Tokens; Healthcare Information Systems

INTRODUCTION

Web applications increasingly support administrative and operational activities across sectors, including healthcare. These applications may provide access to sensitive records, user accounts, administrative functions and service workflows. Their security therefore depends not only on the confidentiality of stored information but also on the strength of the mechanisms used to authenticate users and authorize access. A weakness in an administrator login process can have consequences beyond an individual account because privileged accounts may provide access to functions and information that affect an entire service.

Authentication mechanisms establish whether a user is who they claim to be. A personal identification number (PIN) is a knowledge-based authentication secret. Although PINs can be convenient, short or predictable PINs may be vulnerable to guessing when the application does not impose sufficient complexity requirements or restrict repeated attempts. The OWASP Top 10 identifies identification and authentication failures among the major categories of web application security risk and emphasizes the importance of secure authentication and session management practices (OWASP Foundation, 2021).

The security of authentication does not end when a credential is accepted. Applications commonly issue session identifiers or access tokens to maintain an authenticated state. Bearer tokens require particular care because possession of the token may be sufficient to exercise the privileges associated with it. RFC 6750 explains that any party possessing a bearer token can use it to access protected resources, making secure transmission, storage and handling essential (Jones et al., 2012).

This paper examines a documented authentication weakness identified during an authorized penetration-testing engagement involving the PHC administrator login of MySmartMedic PHC Edition. The case is drawn from a six-month cybersecurity industrial attachment at Uniccon Group of Companies. The wider attachment included security

operations centre activities, Wazuh SIEM administration, FortiGate log integration and authorized web application testing. This article deliberately narrows the analysis to the PIN-based authentication issue because it presents a coherent security case involving credential verification, rate limiting and bearer-token handling.

Statement of the Problem

PIN-based login systems can become vulnerable when applications permit repeated verification attempts without effective throttling and accept weak PIN values. The risk becomes more serious where successful verification leads to the issuance or return of a bearer token that grants access to protected application functions. In the MySmartMedic PHC development environment assessment documented in the industrial training report, testing identified a combination of weak PIN controls, inadequate rate limiting and a response path that returned a valid bearer token after the weak verification condition was satisfied. The case raises a practical concern about the adequacy of authentication controls in web-based healthcare administration platforms.

Aim and Objectives

The aim of this study is to assess the security of a PIN-based administrator authentication mechanism in a healthcare web application through an authorized penetration-testing case study.

The specific objectives are to:

- examine the PIN-verification process used by the assessed administrator login;
- assess whether controls were in place to restrict repeated PIN-verification attempts;
- consider the security implications of the bearer token returned through the documented authentication path; and
- recommend appropriate controls for strengthening PIN authentication and token handling.

Research Questions

- How was the PIN-based administrator authentication mechanism assessed?
- What weaknesses relating to PIN strength and repeated verification attempts were documented?
- What security implications arise when a valid bearer token is returned following weak PIN verification?
- What controls can reduce the identified authentication and token-handling risks?

Significance of the Study

The case provides a practical illustration of how weaknesses in a seemingly simple PIN login process can combine to create a more serious authentication risk. It may assist application developers, security testers and system administrators in recognizing the importance of rate limiting, secure credential verification and token lifecycle controls. It also contributes a Nigeria-based technical case example to discussions of web application security. The findings should, however, be interpreted as evidence from one assessed development environment and not as an estimate of the security condition of healthcare applications generally.

Literature Review

Authentication Security in Web Applications

Authentication is a foundational component of application security because it determines whether a claimant can establish an identity before accessing protected functions. Authentication controls should be designed together with authorization and session-management mechanisms. The OWASP Top 10 highlights identification and authentication failures as a significant web application risk area and recommends robust controls for credential management, authentication flows and session security (OWASP Foundation, 2021).

NIST's digital identity guidance describes requirements and considerations for authenticators, authentication processes and lifecycle management. It emphasizes that authentication assurance depends on the characteristics of the authenticator and the controls surrounding its use (National Institute of Standards and Technology [NIST], 2017). In practical web applications, this means that a login form should not be evaluated only by whether it accepts a correct secret; its resistance to guessing, replay, disclosure and misuse must also be considered.

PIN Complexity and Repeated-Attempt Controls

A PIN is vulnerable when its permitted value space is small, predictable or insufficiently protected against repeated guesses. Complexity requirements can reduce the likelihood of easily guessed values, but complexity alone is not enough. A system that allows unlimited or high-volume attempts may still expose a PIN to automated guessing. Rate

limiting, progressive delays, temporary account lockout where appropriate, risk-based challenges and monitoring of anomalous login patterns are complementary controls.

The OWASP Web Security Testing Guide provides a structured approach to evaluating authentication functionality, including tests of credential policy, authentication bypass and related weaknesses (OWASP Foundation, n.d.-a). Such testing should be conducted only within an explicitly authorized scope and with safeguards that prevent disruption or unauthorized access.

Bearer Tokens and Session Security

Bearer tokens are credentials: a party that possesses a valid token may be able to use it without proving possession of an additional cryptographic key. RFC 6750 therefore treats protection of bearer tokens as a central security requirement and describes risks associated with token disclosure and improper transport (Jones et al., 2012). Applications should issue tokens only after successful authentication, restrict their scope and lifetime, protect them in transit and at rest, and provide mechanisms for revocation or invalidation where required.

A token returned in an authentication response is not automatically a vulnerability; it is often a normal part of a secure authentication protocol. The security concern in this case is the documented relationship between a weak PIN-verification path and the return of a valid token. The critical question is whether an attacker could cause the application to issue or disclose a usable token without satisfying the intended authentication requirements.

Penetration Testing and Risk Management

Penetration testing is a controlled security assessment intended to identify and validate weaknesses in a system within a defined scope. The NIST Risk Management Framework places security assessment within a broader process of identifying, managing and monitoring information-system risks (NIST, 2018). For web applications, a disciplined assessment should document authorization, scope, methodology, evidence, impact and remediation. The OWASP Application Security Verification Standard also provides a basis for specifying and verifying application security requirements (OWASP Foundation, n.d.-b).

Research Gap

General guidance on authentication, application testing and token security is available from standards and security organizations. However, practical case documentation from Nigerian development environments can help demonstrate how multiple weaknesses interact

in a real assessment. This paper addresses that narrow practice-oriented gap by presenting an authorized case study of a healthcare application's PIN-based administrator login. It does not claim to provide a representative survey of Nigerian healthcare systems.

METHODOLOGY

Research Design

The study adopts a descriptive technical case-study design based on an authorized penetration-testing engagement documented in a cybersecurity industrial training report. The case-study approach is appropriate for describing the assessment context, the observed authentication weaknesses and their potential security implications. It is not a controlled experiment and does not involve a survey or statistical testing.

Study Environment and Scope

The assessed system was the development environment of MySmartMedic PHC Edition, specifically the PHC administrator login workflow. The source report identifies the environment as a development instance and records that the engagement was performed as part of authorized security testing under organizational supervision. This paper excludes operational details, credentials, token values, exploit payloads, internal addresses and other sensitive information.

Assessment Procedure

The documented assessment focused on the administrator PIN-verification mechanism. The tester examined the login workflow, observed application responses during authorized verification attempts, considered whether repeated attempts were adequately restricted, and reviewed the authentication response for evidence of token issuance. The resulting finding was documented for organizational review, including a technical explanation, evidence and remediation recommendations.

Tools and Evidence

The wider engagement used a penetration-testing environment that included Kali Linux and Burp Suite for web application assessment. The source report describes the use of these tools in authorized testing activities. This article reports only the high-level assessment outcomes necessary to explain the authentication weakness; it does not reproduce operational instructions or sensitive evidence.

Ethical and Safety Considerations

Testing was conducted within an authorized organizational engagement and under the supervision described in the source report. The assessment is presented in a responsible-disclosure style. No live credentials, bearer-token values, personal records or step-by-step exploitation details are included. Before public submission, the author should obtain written clearance from the system owner and the host organization, confirm permission to name the application, and ensure that screenshots or identifying technical details are removed or approved.

RESULTS

Weakness in PIN Verification

The assessment documented weaknesses in the PIN-based verification process used by the PHC administrator login. In particular, the PIN requirements were described as weak. Weak PIN requirements increase the likelihood that users may select easily guessed values and reduce the effort required to identify a valid PIN when other controls are inadequate.

Inadequate Rate Limiting

The report recorded the absence of effective rate limiting on the PIN-verification endpoint. Without adequate controls on repeated attempts, an attacker may be able to submit repeated guesses at a rate that increases the probability of discovering a valid PIN. The report described a brute-force path during authorized testing. No attempt counts, success rates or time measurements are provided in the source material; accordingly, this paper does not assign numerical estimates to the likelihood of successful guessing.

Bearer-Token Exposure Following Weak Verification

The most consequential observation was that a successful weak-PIN verification path resulted in the application returning a valid bearer token. A bearer token can function as an access credential, so its issuance following an authentication process that does not adequately establish the user's identity may enable unauthorized access to protected functions. The source report characterized the finding as Critical in its internal vulnerability report. This paper reports that classification as the original assessment's rating; it does not independently recalculate a CVSS score because the full scoring vector and supporting details are not reproduced here.

Summary of Findings

Observed issue	Security implication	Recommended control
Weak PIN requirements	Increases the chance of predictable or easily guessed credentials.	Enforce a suitable PIN policy; block common and compromised values where feasible.
Inadequate rate limiting	Allows repeated guesses and may facilitate automated PIN discovery.	Apply server-side throttling, progressive delays, monitoring and risk-based protections.
Bearer token returned after weak verification	May provide an authenticated credential without robust proof of identity.	Issue tokens only after complete authentication; restrict scope and lifetime; protect and revoke tokens.

DISCUSSION

The case illustrates that authentication weaknesses can become more serious when they occur together. Weak PIN requirements increase the predictability of the secret, while inadequate rate limiting can allow repeated attempts. If the application then returns a valid bearer token after a weak verification path, the impact may extend from credential guessing to unauthorized session establishment. This chain demonstrates why authentication should be assessed as an end-to-end process rather than as an isolated login form.

The findings are consistent with established application-security guidance that emphasizes secure authentication, session management and verification of application security requirements (OWASP Foundation, 2021; OWASP Foundation, n.d.-a, n.d.-b). They also reinforce the importance of protecting bearer tokens because possession of such a token may confer access to protected resources (Jones et al., 2012).

The results should be interpreted within the limits of the case. The assessment concerned one development environment and one administrator authentication workflow. The paper does not establish whether the same weakness existed in a production deployment, whether other user roles were affected, or whether the issue remains present after remediation. These questions require confirmation by the system owner and, where appropriate, a controlled retest.

Recommendations

Enforce stronger PIN rules. Set a minimum acceptable length and reject common, sequential, repeated or otherwise easily guessed PINs. The policy should be proportionate to the sensitivity of the account.

Implement server-side rate limiting. Apply per-account and contextual throttling, progressive delays and monitoring for repeated failed attempts. Avoid relying solely on client-side controls.

Use layered authentication for privileged accounts. Consider multi-factor authentication or a stronger authenticator for administrator access, particularly where the account can access sensitive health or operational information.

Control token issuance. A bearer token should be generated and returned only after the complete authentication process has succeeded. Failed, partial or bypassed verification paths must not produce usable credentials.

Protect token lifecycle. Use secure transport, appropriate storage, short and risk-appropriate token lifetimes, narrow scopes, rotation where suitable and revocation or invalidation procedures.

Log and monitor authentication events. Record failed and successful login attempts, suspicious patterns, token issuance events and administrative access, while protecting logs from unauthorized modification and avoiding unnecessary sensitive data.

Retest after remediation. Conduct a documented verification assessment to confirm that the PIN policy, throttling and token-handling controls operate as intended.

Adopt secure development practices. Integrate authentication security requirements and testing into design, implementation, code review and release processes.

CONCLUSION

This paper presented an authorized penetration-testing case study of PIN-based administrator authentication in the development environment of MySmartMedic PHC Edition. The documented assessment identified weak PIN requirements, inadequate rate limiting and a response path that returned a valid bearer token after weak PIN verification. The combination represents a potentially serious authentication weakness because it may allow a party who has not satisfied the intended authentication requirements to obtain an authenticated credential.

The case reinforces the need for strong credential policies, server-side attempt restrictions, secure token issuance and lifecycle management, and independent retesting after remediation. The study is limited to the evidence recorded in one industrial training report and should not be generalized to other healthcare applications without further assessment.

Future work could examine the effectiveness of specific remediation controls through a formally approved retest and compare authentication practices across a wider set of healthcare applications.

REFERENCES

Grassi, P. A., Fenton, J. L., Newton, E. M., Perlner, R. A., Regenscheid, A. R., Burr, W. E., Richer, J. P., Lefkovitz, N. B., Danker, J. M., Choong, Y.-Y., Greene, K. K., & Theofanos, M. F. (2017). *Digital identity guidelines: Authentication and lifecycle management* (NIST Special Publication 800-63B). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63b

Joint Task Force. (2018). *Risk management framework for information systems and organizations: A system life cycle approach for security and privacy* (NIST Special Publication 800-37 Rev. 2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-37r2

Jones, M., & Hardt, D. (2012). *The OAuth 2.0 authorization framework: Bearer token usage* (RFC 6750). RFC Editor. https://doi.org/10.17487/RFC6750

OWASP Foundation. (n.d.-a). *Application security verification standard*. https://asvs.owasp.org/

OWASP Foundation. (n.d.-b). *Web security testing guide*. https://owasp.org/projects/web-security-testing-guide

OWASP Foundation. (2021). *OWASP Top 10: The ten most critical web application security risks*. https://top10.owasp.org/2021/

Uniccon Group of Companies. (2026). *Internal security operations procedures and vulnerability reporting documentation* [Internal organizational documents].