International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)
https://ijcjournal.org/index.php/InternationalJournalOfComputer/index

Methods of Cloud Environment Security Audit in Payment Industry Companies

Njoku Janet Chidinma*

Lagos, Nigeria, Associate, Information security officer (GRC), Multigate Payment Limited
Email: janetnjoku@trolleva.com

Abstract

Cloud infrastructures used by payment companies require audit methods that connect technical control testing with transaction continuity, cardholder data protection, fraud exposure, and regulatory evidence. The article develops an analytical model for assessing cloud environments in payment organizations without experimental measurement or disclosure of internal incidents. The research aim is to classify audit methods suitable for payment companies and explain how technical testing, compliance review, and risk governance form one audit cycle. The source base consists of recent academic studies, payment security reports, and recognized frameworks, including PCI DSS, CSA CCM, CIS Controls, NIST SP 800-53, OWASP WSTG, and ISO/IEC 27017. Comparative analysis, source analysis, typologization, and conceptual synthesis were applied. The article proposes an audit sequence that links cloud configuration review, control mapping, penetration testing, vulnerability assessment, evidence validation, remediation tracking, and GRC reporting. The proposed model gives payment organizations a practical way to align cloud security work with regulated operations.

Keywords: cloud security audit; payment industry; PCI DSS; fintech cybersecurity; vulnerability assessment; penetration testing; cloud compliance; CSA CCM; security governance; operational resilience.

Received: 6/1/2026
Accepted: 8/1/2026
Published: 8/10/2026

---

* *Corresponding author.*

1. Introduction

Payment companies use cloud infrastructure for transaction routing, customer onboarding, fraud monitoring, settlement support, reporting, and application delivery. This dependence creates an audit problem that differs from ordinary enterprise cloud review. A weakness in a payment company's cloud environment affects more than servers, storage, or access settings. Misconfigured storage, excessive privileges, weak segmentation, insecure APIs, or incomplete logs may expose cardholder data, interrupt transaction processing, distort incident evidence, or weaken confidence among customers and regulators.

A payment-oriented cloud audit therefore needs a broader logic than ordinary infrastructure hardening. The audit team has to connect cloud controls with transaction integrity, account data protection, service availability, incident traceability, and regulatory proof. The same IAM error that would be classified as a general access-control weakness in another sector may create a direct route to payment records, administrative consoles, fraud tools, or deployment pipelines in a fintech environment. The same logging gap may prevent a payment company from reconstructing an attack timeline, proving control operation, or responding to a supervisory inquiry.

The aim of the article is to develop an analytical interpretation of cloud environment security audit methods suitable for payment industry companies. The first research objective is to separate payment-sector cloud audit requirements from general cloud security assessment practices. The second objective is to compare the applicability of major methodological frameworks used for cloud audit, compliance review, penetration testing, and vulnerability assessment. The third objective is to formulate an implementation logic that connects technical assessment with governance, remediation, and monitoring.

The novelty of the article lies in treating cloud security audit in payment companies as a connected audit cycle. PCI DSS gives the payment boundary, but it does not replace cloud-native assessment. Penetration testing tests attack paths, but it does not replace compliance validation. GRC review translates findings into ownership and risk decisions, but it does not replace technical evidence. Audit value emerges when the organization sequences these methods around payment data flows, transaction-critical services, third-party dependencies, and post-remediation assurance.

The hypothesis is that cloud environment security audit in payment industry companies gains higher methodological reliability when it integrates regulatory scoping, cloud control assessment, and adversarial testing of transaction-facing systems. A narrow checklist review leaves too much untested because it may confirm that controls exist without proving whether they protect the payment environment under operational pressure.

2. Materials and Methods

The literature search was built around the intersection of cloud security, payment systems, cybersecurity regulation, audit methodology, and application testing. Search queries combined the terms "cloud security audit", "payment systems cybersecurity", "FinTech cybersecurity regulation", "PCI DSS cloud", "cloud control framework", "vulnerability assessment", "penetration testing", "digital payment authentication", and "cloud compliance". Google Scholar, IEEE Xplore, ScienceDirect, Taylor & Francis Online, MDPI, NIST CSRC, PCI

Security Standards Council, Cloud Security Alliance, CIS, OWASP, and ISO catalogue records were checked. The inclusion criteria covered publication between 2020 and 2025 where possible, direct relevance to payment security or cloud audit, methodological usefulness for control assessment, and reliance on recognized academic or professional standards. Works focused only on consumer behaviour, isolated cryptographic protocol design, general IT management, or non-payment cloud adoption were excluded. From an initial pool of more than 40 records, 12 sources were retained. The final corpus covers FinTech cybersecurity challenges [1], payment threat trends [2], payment data security requirements [3], cloud security framework comparison [4], CSA cloud controls Reference [5], CIS security controls [6], NIST security and privacy controls [7], web application testing [8], cloud service security controls [9], digital payment authentication [10], cyber-risk data availability [11], and cloud attack and defence patterns [12].

Comparative analysis was used to distinguish general cloud audit methods from payment-specific audit needs. Source analysis supported the extraction of control logic from academic literature, payment reports, and normative frameworks. Conceptual synthesis connected regulatory scoping, cloud configuration review, penetration testing, vulnerability assessment, and remediation assurance. Typologization classified audit methods by purpose, evidence type, and payment relevance. Analytical generalization supported the formulation of an implementation model suitable for payment organizations without claiming experimental validation.

3. Results

Recent FinTech cybersecurity literature describes payment companies as organizations exposed to a compressed combination of application-layer attacks, identity compromise, infrastructure weaknesses, third-party risk, and regulatory pressure [1]. This position matters for cloud audit because a payment company rarely separates business functionality from security control in a clean way. A checkout API, settlement module, customer wallet, reconciliation service, or fraud monitoring pipeline may sit inside a cloud architecture where identity policies, network segmentation, encryption, key management, logging, and vendor access shape the security posture of the payment product itself.

Ordinary corporate cloud audits often focus on infrastructure hardening, access control, and configuration baselines. Payment-sector audit has to go further. The auditor has to ask whether cloud controls preserve transaction integrity, protect account data, and produce evidence that engineers, compliance officers, incident responders, and management can use. This difference changes the audit question. The question is not limited to whether a control exists. The payment company needs proof that the control protects the specific transaction environment where regulated data and service continuity meet.

Payment threat reporting adds another layer to this distinction. The 2024 payment threat landscape identifies social engineering, malware, distributed attacks, botnets, third-party vendor risks, and monetisation channels as recurring sources of pressure on payment systems [2]. These attacks do not all begin inside cloud infrastructure. Yet cloud environments often host the APIs, dashboards, storage layers, authentication services, and monitoring systems through which impact becomes visible. A fraud campaign may exploit weak customer authentication. A bot attack may stress customer-facing endpoints. A compromised vendor account may touch shared repositories,

logs, support systems, or administrative panels. The cloud audit therefore has to connect technical controls with payment abuse scenarios.

This point affects method selection. A cloud audit limited to static configuration review will miss the link between infrastructure settings and payment abuse. The audit team has to examine detection, logging, privilege boundaries, customer-facing application controls, and third-party integrations. These checks determine whether the organization can contain incidents when fraud and cyberattack patterns move across technical and business layers. The audit method should therefore move from inventory to exposure analysis, then from exposure analysis to evidence and response readiness.

PCI DSS v4.0.1 gives payment-sector audit its most direct regulatory anchor. The standard focuses on environments that store, process, or transmit payment account data and defines technical and operational requirements for protecting that data [3]. In cloud environments, this means the audit begins with scoping. The auditor identifies where account data exists, where it travels, which systems affect the security of the cardholder data environment, and which cloud service provider responsibilities remain outside the payment company's direct control. Scoping affects every later audit step. Weak segmentation expands the audit boundary. Incomplete logging weakens evidence. Encryption without proper key separation creates residual risk even when the encryption control appears present.

Cloud security framework research extends the view beyond payment regulation. Reviews of cloud security frameworks describe shared responsibility, access control, data protection, configuration management, virtualization, monitoring, and compliance alignment as recurring cloud assessment problems [4]. Payment companies usually operate hybrid control stacks. PCI DSS defines what the company has to protect in relation to payment account data. Cloud frameworks explain how cloud-native resources should be assessed across identity, workload, network, storage, logging, and provider boundaries. The audit method must translate payment obligations into inspectable cloud evidence: IAM records, network rules, encryption status, vulnerability reports, container settings, backup protection, and incident response traces.

CSA CCM has a strong place at this translation stage because it provides a cloud-oriented control matrix across multiple domains and supports assessment through mapped control objectives [5]. Its value for payment companies lies in breadth. Auditors may use it to test whether the cloud environment has adequate coverage for identity, infrastructure security, application interfaces, change control, incident management, logging, encryption, business continuity, and supplier governance. Its limits are equally clear. CSA CCM centers on cloud controls. Payment audit still requires data-flow scoping, cardholder data boundary definition, transaction risk interpretation, and evidence that the control set supports regulated payment operations. CSA CCM works best as the cloud control backbone beneath a payment-specific scoping layer.

CIS Controls v8.1 gives the audit a more operational lens. Its safeguards focus on asset inventory, software inventory, secure configuration, account management, access control, vulnerability management, audit logs, malware defenses, data protection, and security awareness [6]. For a payment company, these controls help turn broad security expectations into records that auditors can inspect. The audit team can ask whether cloud assets are

known, whether privileged accounts have owners, whether exposed services have justification, whether vulnerability remediation follows risk priority, and whether logs support investigation of payment-facing events. CIS is useful where the organization needs to move from security intention to operating discipline.

NIST SP 800-53 Rev. 5 expands the audit problem into governance and control management. Its controls cover access control, audit and accountability, configuration management, contingency planning, identification and authentication, incident response, risk assessment, system and communications protection, and supply chain risk management [7]. This breadth gives payment cloud audit two benefits. First, NIST helps translate technical findings into enterprise risk language. Second, it connects cloud control evidence with ownership, policy, remediation responsibility, risk acceptance, and executive reporting. Payment organizations need this bridge because a cloud vulnerability may become a regulatory finding, an operational resilience issue, or a trust failure depending on its location in the transaction chain.

OWASP WSTG assesses another layer of the audit cycle. Payment products commonly expose web applications, APIs, merchant dashboards, administration portals, and customer authentication flows. OWASP WSTG supplies a testing structure for information gathering, configuration and deployment management testing, identity management testing, authentication testing, authorization testing, session management, input validation, error handling, cryptography, business logic, and client-side testing [8]. Its value lies in adversarial verification. A cloud configuration review may confirm that a security group appears restrictive. A web and API test may reveal broken authorization, weak session handling, excessive information disclosure, or business logic flaws that a cloud control matrix does not capture with enough operational depth.

ISO/IEC 27017 adds the shared-responsibility and cloud-service-control perspective [9]. In payment environments, the boundary between customer responsibility and cloud provider responsibility often becomes difficult to prove during audit. Managed databases, serverless functions, hosted identity services, content delivery networks, and managed security tooling distribute security duties across several actors. ISO/IEC 27017 helps structure questions about cloud service customer duties, provider commitments, virtual environment protection, administrative operations, and information handling. Its limitation for payment-sector audit lies in scope. It was not designed around payment account data or transaction processing. It supports cloud governance, while PCI DSS and payment risk modelling define regulated data and service boundaries.

A recent systematic review of multi-factor authentication in digital payment systems places identity assurance at the centre of payment security [10]. This changes how the audit team treats identity. IAM review cannot stop at checking whether administrators use MFA. In payment companies, authentication strength, privilege escalation paths, break-glass accounts, service accounts, API keys, token lifetime, and access to fraud-sensitive systems need joint review. Digital payment authentication studies connect assurance levels with implementation constraints because security controls fail when engineers separate usability, deployment design, and risk classification [10]. For cloud audit, identity testing therefore has to combine configuration review, access-path analysis, and practical verification of privileged actions.

The retained sources support a layered method for cloud audit in payment companies. The first layer defines

payment scope through PCI DSS and payment threat understanding [2,3]. The second layer maps cloud controls through CSA CCM, CIS, NIST, and ISO/IEC 27017 [5-7, 9]. The third layer tests payment-facing applications and identity flows through OWASP methods and authentication assurance logic [8,10]. Academic FinTech cybersecurity literature provides the connecting rationale by showing that FinTech risk combines technology, regulation, customer trust, and operational stability [1]. Cloud security framework research explains why auditors need structured control comparison instead of dependence on one framework [4]. Table 1 presents the resulting audit logic as a conceptual sequence. It does not add numerical data. It adapts the control relationships found in payment security, cloud control, and application testing sources [3,5,7,8].

Table 1: Layered cloud environment security audit cycle for payment industry companies (adapted by the author based on PCI DSS, CSA CCM, NIST SP 800-53, and OWASP WSTG [3,5,7,8])

Audit layer	**Main audit question**	**Typical evidence**	**Payment-sector meaning**
Payment scoping	Which cloud assets affect account data or transaction security?	Data-flow maps, service inventory, segmentation evidence	Defines the regulated audit boundary
Cloud control review	Do cloud controls operate across identity, network, storage, logging, and resilience?	IAM exports, configuration records, encryption settings, monitoring rules	Tests whether cloud operations support secure payment processing
Application and API testing	Can transaction-facing services be abused through technical or business logic weaknesses?	Penetration testing notes, vulnerability reports, OWASP test results	Connects infrastructure posture with real attack paths
Remediation validation	Did teams correct risks and retest them with sufficient proof?	Tickets, retest results, screenshots, change records	Prevents audit closure based only on planned remediation
GRC reporting	Are findings tied to ownership, risk rating, regulatory exposure, and monitoring?	Risk register, control mappings, management reports	Converts technical findings into accountable governance

Table 1 places payment scoping before tool execution. This sequence matters. A scanner can produce hundreds of findings, but the audit team cannot interpret them without knowing which assets affect payment data, transaction integrity, customer authentication, or regulatory evidence. After the boundary is clear, vulnerability assessment, penetration testing, and control review produce evidence that the company can connect to payment risk. This sequencing has particular relevance where a payment organization has faced repeated attacks or pressure

on customer trust. The audit needs to do more than identify isolated weaknesses. It has to show whether the organization can locate exposed assets, test the core product, prioritize remediation, and demonstrate control recovery in language that engineers, managers, and compliance stakeholders understand. The comparison of PCI DSS, CSA CCM, OWASP WSTG, and NIST SP 800-53 clarifies the second research objective. PCI DSS gives payment specificity, but it does not provide a full cloud architecture assessment method [3]. CSA CCM gives cloud breadth, but it does not define cardholder data scoping [5]. OWASP WSTG verifies application and API exposure, but it does not cover cloud governance in full [8]. NIST SP 800-53 supplies control depth and governance linkage, while payment organizations still need PCI DSS to anchor payment data obligations [7]. Framework selection should therefore follow audit function. The audit team should select the tool or framework that answers a defined question, then connect the result to payment impact.

Technical testing requires the same discipline. FinTech cybersecurity studies describe the breadth of cyber risk in digital finance [1]. Payment threat reports show that attack patterns often combine social, technical, and third-party elements [2]. OWASP WSTG gives structure for testing application weaknesses [8]. Digital payment authentication research highlights assurance gaps in identity implementations [10]. These sources point to one practical conclusion: penetration testing and vulnerability assessment should operate inside a payment-risk model. Their output should be mapped to control failure, payment impact, and remediation ownership. A vulnerability without payment relevance may still require correction, but weaknesses in authorization, session management, API exposure, or privileged cloud access deserve faster escalation because they may affect transaction trust.

GRC enters the audit before the final report. CIS Controls translate expectations into operational safeguards [6]. NIST SP 800-53 connects controls with organizational risk management [7]. ISO/IEC 27017 clarifies cloud service responsibilities [9]. CSA CCM maps cloud control domains [5]. In a payment company, GRC should shape scoping rules, evidence requirements, control ownership, and remediation acceptance from the start. If the organization waits until the technical report is finished, it may repair visible vulnerabilities while leaving responsibility, recurring monitoring, regulatory reporting, and control durability unresolved.

4. Discussion

A payment cloud audit should be treated as disciplined evidence work. Payment companies often operate under compressed release cycles, multiple service providers, cloud-native deployment pipelines, and strict availability expectations. An audit model that disrupts operations or produces findings without ownership will lose force. A stronger approach starts with a practical boundary question: which cloud assets affect payment data, transaction integrity, service availability, or customer trust? This boundary allows the audit team to select methods with less waste and greater precision.

A workable implementation model begins with asset and data-flow reconstruction. The audit team documents cloud accounts, subscriptions, virtual networks, managed databases, object storage, container clusters, serverless functions, identity providers, CI/CD pipelines, secrets stores, monitoring services, and third-party connections. The second step defines payment scope. Systems are classified by their relation to cardholder data, transaction processing, administration, fraud monitoring, settlement support, and reporting. The third step maps controls. PCI

DSS sets the payment baseline. CSA CCM and ISO/IEC 27017 refine the cloud control vocabulary. CIS supports operational checks. NIST links evidence to risk governance. The fourth step verifies exposure through vulnerability assessment, penetration testing, and OWASP-based application testing. The fifth step checks remediation through retesting and evidence review. The sixth step transfers residual findings into GRC records, control exceptions, risk acceptances, and monitoring requirements. Table 2 compares the audit functions of the main frameworks. The table does not rank them. It shows where each framework fits inside a payment-oriented cloud audit.

Table 2: Applicability of security frameworks in cloud environment audit for payment industry companies (compiled by the author based on [3, 5-9])

Framework or standard	**Strongest audit function**	**Best use in payment cloud audit**	**Main limitation**
PCI DSS v4.0.1	Payment account data protection and compliance validation	Defines scope, regulated evidence, segmentation, logging, vulnerability management, and control obligations for payment data environments	Does not provide a complete cloud architecture assessment method
CSA CCM	Cloud control mapping across cloud domains	Structures cloud security assessment across identity, infrastructure, logging, encryption, resilience, and provider responsibility	Requires payment-specific interpretation
CIS Controls v8.1	Operational security safeguards	Supports practical checks for inventory, configuration, access, vulnerability management, logging, and incident readiness	Less specific to payment data scoping
NIST SP 800-53 Rev. 5	Control catalogue and governance linkage	Connects technical findings with risk management, accountability, contingency planning, supply chain risk, and audit evidence	Broad scope requires tailoring for payment environments
OWASP WSTG	Web application and API testing	Guides penetration testing of payment-facing portals, APIs, authentication, authorization, session handling, and business logic	Does not assess full cloud governance
ISO/IEC 27017	Cloud service security guidance	Clarifies cloud customer and provider control responsibilities	Older standard and not payment-specific

The table supports a combined-framework strategy. A payment organization gains little from framework adoption if the audit team cannot tie each framework to evidence. PCI DSS scoping should lead to data-flow records and segmentation proof. CSA CCM should lead to cloud control documentation. CIS should lead to operational proof. OWASP should lead to reproducible testing results. NIST should lead to risk ownership and executive visibility. ISO/IEC 27017 should clarify which actor controls each part of the cloud service chain.

The findings extend previous research by connecting separate lines of cloud security analysis within one payment-oriented audit sequence. AlBenJasim and his colleagues [1] describe the regulatory and cybersecurity pressures affecting FinTech companies but do not formulate an integrated cloud audit procedure. Chauhan and Shiaeles [4] compare cloud security frameworks without linking them to payment data scoping and transaction exposure. Tran-Truong and his colleagues [10] examine authentication in digital payments, while Cremer and his colleagues [11] identify limitations in cyber-risk data availability and Dawood and his colleagues [12] review cloud attack and defence methods. The present study connects these findings by assigning each framework and testing method to a defined audit function. PCI DSS establishes the payment boundary, cloud frameworks structure control assessment, OWASP supports adversarial testing, and NIST links findings with ownership and remediation. The contribution therefore lies in the sequence of evidence collection rather than in proposing another control catalogue.

The author's position is that payment cloud audits should be designed around failure paths. A failure path connects a technical weakness with payment impact. Excessive privilege in a cloud account may open access to logs, transaction data, deployment pipelines, or fraud controls. An exposed storage bucket may leak evidence, customer data, reconciliation files, or operational reports. Weak API authorization may enable transaction manipulation, account takeover, or merchant-level abuse. This way of reading findings prevents audit reports from becoming lists of disconnected technical items. Table 3 proposes monitoring metrics for the period after audit closure. The purpose is to connect remediation with sustained control operation. The metrics are qualitative and managerial because the article does not use internal datasets or invented numerical evidence.

Table 3: Monitoring metrics for post-audit control durability in payment cloud environments (compiled by the author based on [3, 6-9,11,12])

Monitoring area	**Metric type**	**Audit question after remediation**	**Practical use**
Asset visibility	Coverage metric	Are all payment-relevant cloud assets visible in the inventory?	Detects shadow services and untracked infrastructure
Identity governance	Control operation metric	Are privileged accounts reviewed, justified, and protected?	Reduces excessive access and unmanaged administrator risk
Vulnerability	Remediation	Are high-risk findings assigned,	Prevents report closure

management	discipline metric	corrected, retested, and closed with evidence?	without technical proof
Logging and detection	Evidence readiness metric	Can payment-facing events be traced across cloud, application, and identity layers?	Supports incident investigation and audit defensibility
Segmentation	Boundary integrity metric	Do cloud network and access controls preserve the payment data boundary?	Prevents scope expansion and lateral movement
Application testing	Exposure metric	Are APIs, authentication flows, authorization rules, and business logic tested before major releases?	Connects product security with release governance
Third-party access	Supplier risk metric	Are provider access, support channels, integrations, and inherited controls documented?	Clarifies shared responsibility
GRC reporting	Accountability metric	Are findings linked to risk owners, deadlines, residual risk decisions, and control exceptions?	Converts technical remediation into governance action

The table addresses a common weakness in cloud audits. Teams often treat audit closure as the end of security work. In cloud environments, closure lasts only until the next deployment, integration, identity change, region expansion, or managed-service update. Monitoring should therefore focus on drift. The most dangerous post-audit condition appears when the organization loses visibility into whether a corrected control still operates. A patched vulnerability leaves a clear record. A drifting identity rule, missing log source, or undocumented service creates uncertainty at the exact point where payment companies need evidence.

Audit implementation should follow risk concentration. Systems directly connected to payment data, authorization decisions, transaction routing, reconciliation, or customer authentication receive the highest audit attention. Systems with administrative access to those components require similar treatment, even if they do not process payment data directly. Development and CI/CD environments need separate scrutiny because they create indirect routes into production. Monitoring platforms, logging stores, secrets managers, and backup systems require protection because they hold control evidence or recovery capability. Stakeholder communication affects audit quality. Technical teams need precise findings, reproduction steps, affected assets, and remediation expectations. Compliance teams need control mapping, evidence status, and regulatory exposure. Management needs risk language, business impact, owners, and a remediation sequence. A payment cloud audit fails when it speaks only to one audience. The report should therefore have layered communication: a technical annex for engineers, a control matrix for compliance, and a risk summary for executives.

The study has several limitations. It is based on conceptual synthesis rather than empirical testing and does not use internal audit records, incident datasets, penetration-testing results, or company-level measurements. The source corpus is limited to 12 academic publications, professional reports, and standards selected for direct methodological relevance. The proposed audit cycle was not tested in payment companies differing in size, jurisdiction, cloud architecture, or business model. The study does not estimate implementation cost, audit duration, staffing requirements, remediation speed, or changes in incident frequency. The proposed model therefore requires further validation through case studies and longitudinal analysis of audit results.

A mature payment cloud audit should be evaluated through four outcomes. The first is boundary clarity: the organization knows which cloud systems affect regulated payment operations. The second is evidence reliability: teams can prove control operation without improvised documentation. The third is remediation quality: engineers correct findings and retest them. The fourth is governance continuity: residual risks remain visible after the audit team leaves. For payment companies, this continuity protects more than technical infrastructure. It protects the organization's ability to explain, defend, and improve its security posture under operational and regulatory pressure.

5. Conclusion

Cloud environment security audit in payment industry companies differs from general cloud assessment because the audited infrastructure supports regulated data flows, transaction continuity, fraud-sensitive operations, and customer trust. The payment sector requires scoping that links cloud assets with account data, transaction-facing applications, privileged administration, provider responsibility, and operational resilience. A generic cloud checklist leaves gaps because it cannot connect technical evidence with payment data boundaries and transaction risk.

The comparison of methodological frameworks confirms the need for functional separation. PCI DSS defines the payment compliance boundary. CSA CCM structures cloud control coverage. CIS Controls support operational inspection. NIST SP 800-53 connects technical controls with risk governance. OWASP WSTG verifies application and API exposure. ISO/IEC 27017 clarifies cloud-service responsibility. Each framework contributes a distinct audit function, while none covers the full payment cloud audit cycle alone.

The proposed implementation model links asset discovery, payment scoping, control mapping, vulnerability assessment, penetration testing, remediation validation, and GRC reporting. The comparison conceptually supports the hypothesis that cloud environment security audit gains greater methodological reliability when regulatory scoping, cloud control assessment, and adversarial testing are integrated within one audit cycle. Empirical confirmation requires company audit records, testing results, incident data, and post-audit monitoring.

References

[1]. AlBenJasim, S., Dargahi, T., Takruri, H., & Al-Zaidi, R. (2024). Fintech cybersecurity challenges and regulations: Bahrain case study. *Journal of Computer Information Systems*, *64*(6), 835–851. https://doi.org/10.1080/08874417.2023.2251455

[2]. European Payments Council. (2024). *2024 payment threats and fraud trends report* (EPC162-24, Version 1.0). https://www.europeanpaymentscouncil.eu/sites/default/files/kb/file/2024-12/EPC162-24%20v1.0%202024%20Payments%20Threats%20and%20Fraud%20Trends%20Report_0.pdf

[3]. PCI Security Standards Council. (2024). *Payment Card Industry Data Security Standard: Requirements and testing procedures* (Version 4.0.1). https://www.pcisecuritystandards.org/document_library

[4]. Chauhan, M., & Shiaeles, S. (2023). An analysis of cloud security frameworks, problems and proposed solutions. *Network, 3*(3), 422–450. https://doi.org/10.3390/network3030018

[5]. Cloud Security Alliance. (2021). *Cloud controls matrix* (Version 4.0). https://cloudsecurityalliance.org/artifacts/cloud-controls-matrix-v4

[6]. Center for Internet Security. (2024). *CIS critical security controls* (Version 8.1). https://www.cisecurity.org/controls/v8-1

[7]. Joint Task Force Interagency Working Group. (2020). *Security and privacy controls for information systems and organizations*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-53r5

[8]. OWASP Foundation. (n.d.). *Web security testing guide* (Version 4.2). https://owasp.org/www-project-web-security-testing-guide/stable/

[9]. ISO/IEC. (2015). *ISO/IEC 27017:2015 information technology—security techniques—code of practice for information security controls based on ISO/IEC 27002 for cloud services*. https://www.iso.org/standard/43757.html

[10]. Tran-Truong, P. T., Pham, M. Q., Son, H. X., Nguyen, D. L. T., Nguyen, M. B., Tran, K. L., Van, L. C. P., Le, K. T., Vo, K. H., Kim, N. N. T., Nguyen, T. M., & Nguyen, A. T. (2025). A systematic review of multi-factor authentication in digital payment systems: NIST standards alignment and industry implementation analysis. *Journal of Systems Architecture, 162*, 103402. https://doi.org/10.1016/j.sysarc.2025.103402

[11]. Cremer, F., Sheehan, B., Fortmann, M., Kia, A. N., Mullins, M., Murphy, F., & Materne, S. (2022). Cyber risk and cybersecurity: A systematic review of data availability. *The Geneva Papers on Risk and Insurance - Issues and Practice, 47*(3), 698–736. https://doi.org/10.1057/s41288-022-00266-6

[12]. Dawood, M., Tu, S., Xiao, C., Alasmary, H., Waqas, M., & Rehman, S. U. (2023). Cyberattacks and security of cloud computing: A complete guideline. *Symmetry, 15*(11), 1981. https://doi.org/10.3390/sym15111981