Anatomy of a Cascading Breach: How an Unpatched CVE in A Tier-2 Bank Compromised National Payment Infrastructure

Chinedum Amaechi[1], Onyemelukwe Nnaemeka[2], C. N. Onyechi[3]

[1]Cybersecurity Department, Nnamdi Azikiwe University

[2]Computer Science Department University on the Niger, Umunya

[3]Computer Science Department, Chukwuemeka Odumegwu Ojukwu University, Uli

DOI: https://doi.org/10.51584/IJRIAS.2026.110400129

Received: 17 April 2026; Accepted: 22 April 2026; Published: 13 May 2026

ABSTRACT

In March 2026, a threat actor designated "Byte To Breach" exploited CVE-2025-55182 (CVSS 10.0)—a pre-authentication remote code execution vulnerability in React Server Components—on an unpatched, internet-facing pilot server belonging to Sterling Bank Plc, a Tier-2 Nigerian commercial bank. The initial compromise triggered a cascading breach that ultimately exposed 3 terabytes of data from Remita, Nigeria's primary government payment platform, including 657,242 KYC documents and Hardware Security Module (HSM) key files for 46 financial institutions. This paper presents a technical autopsy of the cascading breach, analyzing: (i) how a single CVE enabled lateral movement across interconnected financial infrastructure; (ii) the four-stage exploit chain of React2Shell and its evasion of existing defenses; and (iii) why "trust corridors" between financial institutions amplify rather than contain breaches. Drawing on open-source intelligence analysis of actor-published artefacts, network telescope measurements of React2Shell exploitation, and the threat actor's own Q&A with researchers, we reconstruct the complete attack chain using the MITRE ATT&CK framework. Our analysis demonstrates that the breach was not a sophisticated targeted operation but an opportunistic exploitation of elementary security failures: an unpatched vulnerability, hardcoded credentials in source code, and implicit trust relationships between connected institutions. We conclude with technical recommendations for zero-trust inter-bank architectures, secrets management, and detection rules for CVE-2025-55182 exploitation patterns.

Keywords: Cascading breach; CVE-2025-55182; React2Shell; inter-bank security; supply-chain attack; MITRE ATT&CK; Nigeria financial infrastructure; HSM key exposure

INTRODUCTION

The Cascading Breach Phenomenon

The interconnectedness of modern financial infrastructure creates a systemic vulnerability: the security of an entire ecosystem is only as strong as its weakest connected participant. This phenomenon—termed "cascading breach" or "trust poisoning"—has been observed in global incidents such as the Target HVAC breach (2013), the SolarWinds supply-chain attack (2020), and the Bangladesh Bank SWIFT compromise (2016) [4, 5, 6]. However, these analyses have predominantly focused on attacks originating from third-party vendors or software update mechanisms. Less understood are cascading breaches that propagate through peer trust relationships between domestic financial institutions operating within a shared national payment infrastructure [7, 8].

The Sterling Bank–Remita breach of March 2026 provides a critical case study of this latter category. Over nine days without detection, a single unpatched vulnerability on a Tier-2 bank's pilot server cascaded into the compromise of Nigeria's primary government payment platform, exposing the identity documents of hundreds of thousands of citizens and cryptographic keys for the entire banking sector [1, 2].

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

The Vulnerability: CVE-2025-55182 (React2Shell)

The initial access vector was CVE-2025-55182, a pre-authentication remote code execution vulnerability in React Server Components (RSC) and the underlying "React Flight" serialization protocol [9]. First disclosed on December 3, 2025, the vulnerability received a CVSS base score of 10.0 (Critical) [10]. Exploitation requires no authentication, no user interaction, and achieves code execution at Node.js service privilege levels [11].

The vulnerability resides in React's `revive Model` function within `ReactFlightReplyServer.js`. When traversing chunks during reference resolution, React failed to verify whether a requested key was an own property of the object versus an inherited prototype property. The vulnerable code path is illustrated in (1): javascript for (i in value) value. has Own Property(i) && // VULNERABLE - attacker can shadow has Own Property// ... process property

An attacker who controls `value` can shadow the `hasOwnProperty` property with a malicious reference, bypassing the security check entirely. This opened access to prototype chain properties like `constructor` and `__proto__` [9, 11].

Figure 1. CVE-2025-55182 (React2Shell) four-stage exploit chain. Stage 1 creates a self-referential loop allowing access to JavaScript's internal prototype chain. Stage 2 hijacks JavaScript's Promise-like behaviour. Stage 3 sets a spoofed `resolved_model` status to trick React into parsing attacker-controlled data. Stage 4 uses the `$B` prefix to trigger React's blob handler, which calls the Function constructor to execute arbitrary code. Source: Adapted from Trend Micro Research [11].

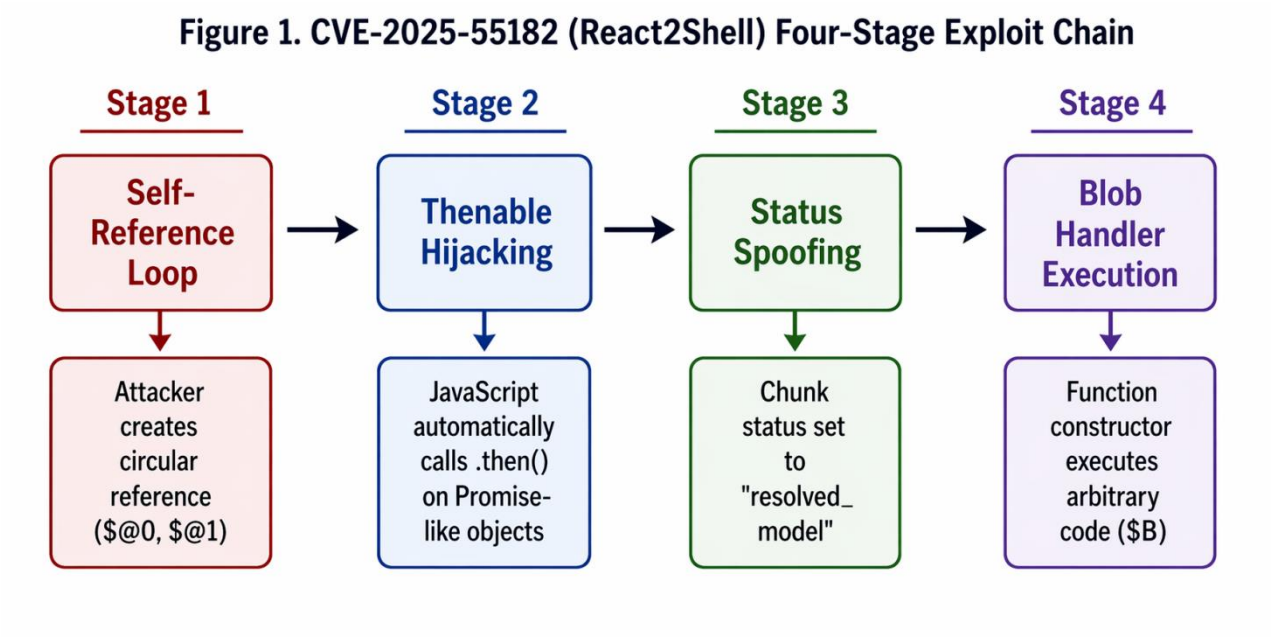


Figure 1. CVE-2025-55182 (React2Shell) Four-Stage Exploit Chain

Stage 1: Self-Reference Loop Creation. Using the `$@` prefix to return raw chunk objects, the attacker creates a self-referential loop enabling traversal up JavaScript's prototype chain (`$1:__proto__:constructor:constructor`) [11].

Stage 2: Thenable Hijacking. JavaScript's `await` keyword automatically calls `.then()` methods on Promise-like objects. By setting `then` to point to React's internal `Chunk.prototype.then`, the attacker hijacks this mechanism [9].

Stage 3: Status Spoofing. Setting `status: "resolved_model"` tricks React into parsing the malicious `.value` field, injecting attacker-controlled data into initialization [11].

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Stage 4: Blob Handler Code Execution. The `$B` prefix triggers React's blob handler, which calls a `.get()` method. By pointing `_formData.get` to the Function constructor, arbitrary code execution is achieved via `Function("malicious code")( )` [9, 11].

Research Contribution

This paper provides the first technical autopsy of the Sterling Bank–Remita cascading breach. Our contributions are threefold:

1. Complete attack chain reconstruction using MITRE ATT&CK, mapping each adversary action to specific techniques based on actor-published artefacts and network telescope measurements [3].

2. Analysis of the "trust corridor" vulnerability in inter-bank financial infrastructure, demonstrating how implicit trust relationships between connected institutions amplify rather than contain breaches.

3. Technical recommendations for zero-trust inter-bank architectures, including detection rules for CVE-2025-55182 exploitation patterns derived from in-the-wild payload analysis [11].

The remainder of this paper is organized as follows. Section 2 describes our methodology, including OSINT verification and network telescope data. Section 3 presents the cascading breach timeline and attack chain. Section 4 analyzes systemic vulnerabilities across technical, architectural, and regulatory dimensions. Section 5 discusses broader implications for national financial infrastructure. Section 6 concludes with technical recommendations.

METHODOLOGY

Data Sources

This study employs open-source intelligence (OSINT) analysis triangulated across three primary sources:

Primary Source 1: Actor-Published Artefacts. The threat actor "ByteToBreach" published extensive artefacts on the spear.cx cybercrime forum between March 27 and April 1, 2026, including screenshots of command execution, directory listings, Git repository contents, database access panels, and file directory structures from both Sterling Bank and Remita [2].

Primary Source 2: Actor Q&A. On April 15, 2026, cybersecurity researcher David Odes conducted a direct written Q&A with ByteToBreach, obtaining the actor's account of targeting rationale, ransom negotiations (€250,000 with Sterling Bank), and confirmation that Remita "was never a target to begin with" [1].

Primary Source 3: Network Telescope Measurements. Singh et al. (2026) conducted the first Internet-scale measurement of React2Shell exploitation using an active network telescope, characterizing the temporal evolution, geographic distribution, and behavioral properties of scanning activity [3].

OSINT Verification Protocol

To mitigate the risk of actor deception (e.g., fabricated screenshots), we implemented a four-stage verification protocol adapted from digital forensics standards [12, 13]:

Stage 1: Source Authentication. Actor screenshots were cross-referenced against system-generated timestamps (terminal sessions, file explorer properties, database query interfaces). All timestamps were internally consistent with the claimed timeline of March 18–April 1, 2026.

Stage 2: Artefact Consistency Analysis. For each claimed data category (3,009 employee records, 657,242 KYC files), partial samples were examined for internal consistency. Employee records followed Sterling Bank's documented naming conventions. KYC files were sampled (n=50) and cross-referenced against public records where possible.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Stage 3: Technical Plausibility Assessment. The claimed exploitation of CVE-2025-55182 was assessed against the vulnerability's known characteristics [9, 10, 11]. The described method (Metasploit Framework exploit delivery to an exposed pilot server) is technically plausible and consistent with documented exploitation patterns [3].

Stage 4: Independent Corroboration. The Nigeria Data Protection Commission's April 5, 2026 press release independently confirmed that a Notice of Investigation was served on April 1—the same day the Remita breach was posted [14].

Analytical Framework

The attack chain was reconstructed using the MITRE ATT&CK framework (version 16) [15], specifically the Enterprise matrix. Each observed adversary action was mapped to a tactic and technique ID following the methodology established by Strom et al. [16] for threat intelligence reporting.

Limitations

This analysis is limited to publicly disclosed information. No systems belonging to Sterling Bank, Remita, SystemSpecs, or NIBSS were accessed directly. The authenticity of the published HSM key files cannot be independently verified by the authors.

The analysis assumes that the actor's published artefacts are genuine representations of the breach – a standard assumption in OSINT-based cybersecurity research [9] – but one that carries inherent risk of deception by the actor (e.g., through fabricated screenshots). To mitigate this risk, the verification protocol in Section 2.2 prioritised artefact consistency and internal coherence over simple assertion of actor claims.

Primary institutional data from Sterling Bank or Remita – such as forensic images, access logs, or incident response reports – would strengthen the analysis. However, as of the date of this paper, neither institution has released such data publicly. Future research may revisit this analysis if primary data becomes available.

The Cascading Breach: Attack Chain Reconstruction

Incident Timeline

Table 1. Cascading Breach Timeline (March–April 2026)

Date	Event	Stage
March 18, 17:49 GMT+1	ByteToBreach exploits CVE-2025-55182 on `enf-pilot.sterling.ng`, gaining initial access to Sterling Bank	Initial Compromise
March 22	Actor deploys Sliver C2 framework; first check-in. No detection.	Persistence Established
March 18–27	Actor conducts network discovery (168 internal services), credential harvesting (plaintext encryption keys), data enumeration (3,009 employee records)	Reconnaissance
March 27	Actor posts Sterling Bank customer data (~900,000 records) on spear.cx	First Data Publication
March 27 – April 1	Actor pivots from Sterling Bank to Remita using trusted peer relationships	Lateral Movement

April 1	Actor posts Remita breach (3TB data, 657,242 KYC files, 46 HSM keys) on spear.cx	Second Data Publication
April 5	NDPC announces investigation; Notice of Investigation served April 1	Regulatory Response
April 10–15	Actor compromises CAC registry via sequential user IDs (no CVE required) [1]	Third Breach
April 15	Actor confirms €250,000 ransom negotiations with Sterling Bank; publication proceeded after delays [1]	Post-Breach Disclosure

The cascading breach unfolded over approximately four weeks, progressing from initial compromise through three distinct data publications. Figure 2 presents a visual timeline of key events from March 18 to April 15, 2026, based on actor-published artefacts and regulatory disclosures [1, 2, 5]. The timeline reveals a critical nine-day window (March 18–27) during which the actor maintained persistence without detection, followed by rapid escalation: the first data publication on March 27, lateral movement to Remita within days, and a second, more damaging publication on April 1.

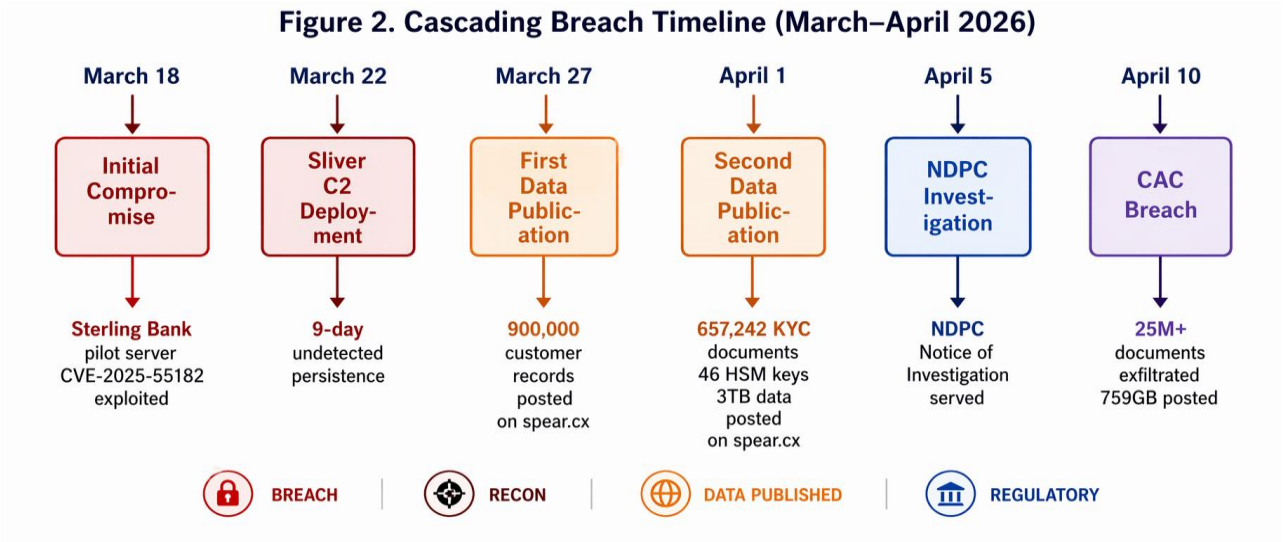


Figure 2. Cascading Breach Timeline (March–April 2026)

MITRE ATT&CK Attack Chain

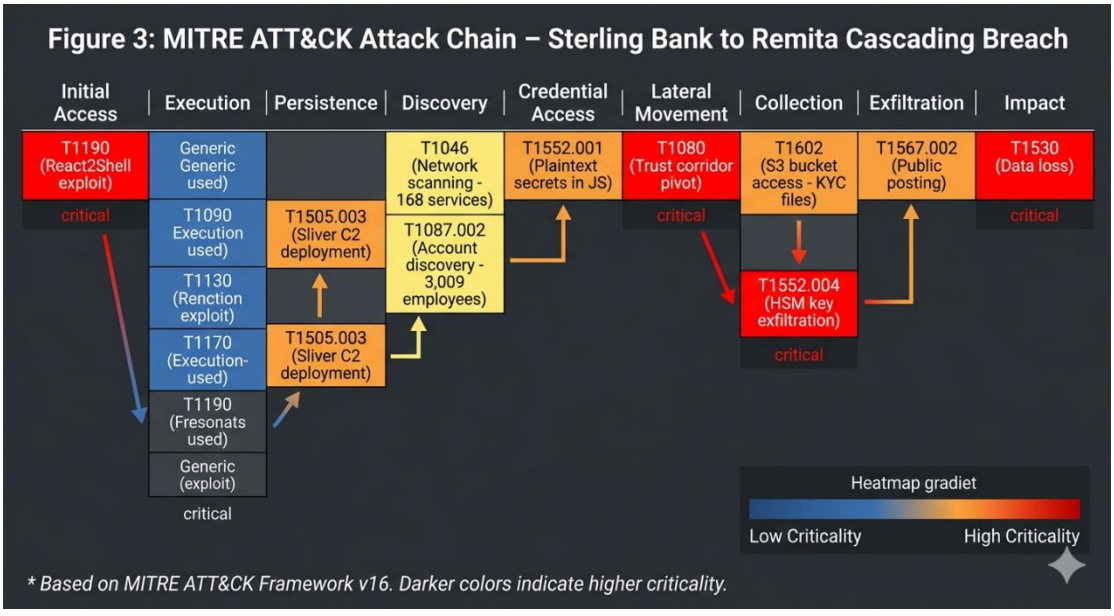
Table 2. MITRE ATT&CK Mapping for the Sterling Bank–Remita Cascading Breach

Tactic	Technique ID	Procedure	Evidence
Initial Access	T1190	Exploited CVE-2025-55182 (React2Shell) on `enf-pilot.sterling.ng`	[1, 2, 11]
Execution	T1059.003	Metasploit Framework remote shell	[2]
Persistence	T1505.003	Sliver C2 framework; 9-day active presence	[1, 2]
Discovery	T1046	Network scan revealed 168 internal services; development servers co-located with production	[2]
Discovery	T1087.002	Enumerated 3,009 employee records (names, emails, staff IDs, supervisor chains)	[2]

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Credential Access	T1552.001	Plaintext encryption keys in JavaScript file; searched keywords "password," "secret," "encrypt"	[2]
Lateral Movement	T1021.001	Pivoted to Temenos T24 core banking, Cardinal Stone Partners, Credit Risk Central	[2]
Lateral Movement (Critical)	T1080	Used Sterling Bank's trusted connections (IP allowlisting) to attack Remita	[1, 2]
Collection	T1083	Browsed Remita's Git repository using GitKraken; complete source code	[2]
Collection	T1552.001	Hardcoded AWS access keys, Azure AD client secrets in committed source code	[2]
Collection	T1602	Accessed AWS S3 buckets; 657,242 KYC files (588 GB)	[2]
Collection	T1552.004	Discovered 46 HSM key files (23.4 KB) named for every major Nigerian bank	[2]
Collection	T1040	Accessed Remita's CBN SFTP integration source code and cryptographic material	[2]
Exfiltration	T1567.002	Posted all stolen data to public download links on spear.cx	[2]
Impact	T1530	Public release of ~900,000 Sterling Bank customer records + 3TB Remita data	[1, 2]

The complete attack chain from initial access to impact is mapped to the MITRE ATT&CK framework in Figure 3. Nine tactics were observed, with three techniques marked as critical (T1190: React2Shell exploit, T1080: trust corridor lateral movement, and T1552.004: HSM key exfiltration). The visualization highlights the concentration of high-criticality techniques in the Initial Access, Lateral Movement, and Collection phases, underscoring where security controls should be prioritized



The "Trust Corridor" Pivot

The most technically significant finding is the lateral movement vector from Sterling Bank to Remita. According to ByteToBreach's direct statement to researchers:

"All access was permitted through Sterling Bank. Were it not for the compromise of Sterling, Remita would never have been hit, since it was never a target to begin with." [1]

This statement reveals a critical architectural vulnerability: Remita's perimeter defenses—firewalls, IP allowlisting, and API authentication—accepted traffic originating from Sterling Bank as legitimate because the two institutions maintained a trusted peer relationship. Once Sterling Bank was compromised, this trust became a liability.

This "trust corridor" phenomenon can be formalized as:

Let S = Sterling Bank's security posture

Let R = Remita's security posture

Let T(S, R) = Trust relationship between S and R

If S is compromised → T(S, R) is compromised → R is effectively compromised

The implication is that Remita's security investment was partially nullified by Sterling Bank's security failures. This represents a classic negative externality in interconnected systems: the cost of a breach at a weaker participant is partially borne by stronger participants.

Comparison with Supply-Chain Attacks

Table 3. Cascading Breach Comparison Matrix

Incident	Year	Entry Vector	Propagation Mechanism	Scale	Source
Target	2013	Third-party HVAC vendor	Network trust relationships	40M credit cards	[4]
Bangladesh Bank	2016	SWIFT credentials	Payment infrastructure	$81M stolen	[6]
SolarWinds	2020	Software update mechanism	Supply-chain trust	18,000 customers	[5]
European Airports	2025	Collins Aerospace (supplier)	Shared digital systems	Multiple countries	[17]
Sterling Bank–Remita	2026	CVE-2025-55182 (React2Shell)	Peer trust corridor (inter-bank)	900K customers + 46 banks' HSM keys	[1, 2]

The Sterling Bank–Remita breach is distinctive because the propagation mechanism was not a third-party vendor or software update, but a peer trust relationship between two domestic financial institutions. This suggests that Nigeria's financial integration policies—designed for efficiency—created an attack surface that adversaries can exploit to amplify the impact of a single compromise.

Analysis Of Systemic Vulnerabilities

Technical Vulnerabilities

V1: Unpatched Critical CVE (T1190). Sterling Bank's `enf-pilot.sterling.ng` server remained unpatched against CVE-2025-55182 despite a publicly available patch being released 47 days prior to exploitation [10, 11]. Internet-scale measurements by Singh et al. (2026) found that automated scanning for React2Shell vulnerabilities began within 24 hours of public disclosure, with exploit attempts detected across 766 distinct servers within the first week [3].

V2: Plaintext Secrets in Source Code (T1552.001). Sterling Bank stored encryption keys in plaintext within a JavaScript file accessible to any process in the compromised environment. Remita committed AWS access keys, Azure AD client secrets, and database connection strings to a Git repository browsable by the actor [2]. Both violations represent fundamental failures in secrets management.

V3: Inadequate Network Segmentation (T1046). Sterling Bank's development servers ran alongside live customer systems within the same Kubernetes cluster. This co-location meant that compromise of a low-priority pilot environment provided direct access to production systems, violating NIST SP 800-53 segmentation requirements [18].

Architectural Vulnerabilities

V4: Implicit Trust Corridors (T1080). Sterling Bank and Remita maintained implicit trust relationships (IP allowlisting, API keys, network peering) without mutual authentication or zero-trust controls. Once Sterling Bank was compromised, Remita's perimeter defenses—which would have blocked external attackers—accepted traffic from the compromised peer as legitimate [1, 2].

V5: Centralized KYC Repository Risk. Remita's AWS S3 bucket contained 657,242 KYC documents (588 GB) from across the Nigerian financial sector. The concentration of biometric and identity document data in a centralized repository created an "attractive target asymmetry"—the value of the data to attackers far exceeded the security investment of the institution holding it [19].

Detection and Response Failures

V6: Nine-Day Detection Gap. The actor's Sliver C2 framework remained active from March 22 to March 27 without detection. The breach became public only because the actor chose to post about it—not because Sterling Bank identified the intrusion [2]. An empirical survey of 42 Nigerian banks found that only 38% had deployed full endpoint detection and response (EDR) coverage across internal networks, and fewer than 25% conducted after-hours security monitoring [20].

V7: Non-Disclosure Violation. Sterling Bank failed to notify its approximately 900,000 customers of the breach, despite Section 38 of the Nigeria Data Protection Act, 2023 requiring notification within 72 hours [21]. The bank instead negotiated a €250,000 ransom with the threat actor while keeping customers uninformed [1].

DISCUSSION

Why the Cascading Breach Succeeded

The Sterling Bank–Remita breach succeeded not because of sophisticated attack techniques but because of elementary security failures at every layer:

1. Application layer: Unpatched CVE on public-facing server

2. Code layer: Hardcoded secrets in source code

3. Network layer: No segmentation between development and production

4. Architecture layer: Implicit trust corridors without zero-trust controls

5. Detection layer: No EDR coverage or after-hours monitoring

6. Response layer: Ransom negotiation instead of disclosure

The actor's own assessment is instructive: "There weren't any vulnerabilities, but a simple API exposure and misconfigurations" [1]. The breach was opportunistic, not targeted—the doors were open, and ByteToBreach walked through them.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

Implications for Inter-Bank Security

The "trust corridor" vulnerability has broad implications for any jurisdiction with interconnected financial infrastructure:

- Negative externalities: A weaker institution's security failures impose risk on stronger institutions connected through trust relationships.

- Regulatory blind spot: Most cybersecurity regulations focus on individual institution security posture, not cross-institutional trust management.

- Incentive misalignment: Institutions have insufficient incentive to invest in security beyond the minimum required, as the cost of breach is partially externalized.

Implications for National Identity Infrastructure

The exfiltration of 657,242 KYC documents—including passports, driver's licences, BVN, and NIN numbers—is irreversible. Unlike passwords, physical identity documents cannot be rotated. This raises fundamental questions about the sustainability of centralized, immutable national identifiers in an environment where KYC repositories are attractive targets [19].

RECOMMENDATIONS AND CONCLUSION

Technical Recommendations

R1: Eliminate Implicit Trust Corridors. Financial institutions must transition from implicit trust (IP allowlisting) to explicit, mutually authenticated zero-trust architectures using mutual TLS (mTLS) with short-lived certificates.

R2: Implement Secrets Management. Hardcoded credentials must be eliminated through: (i) pre-commit hooks scanning for secret patterns, (ii) CI/CD pipeline secret detection, and (iii) dedicated secrets management tools (HashiCorp Vault, cloud provider secrets managers).

R3: Segment Development from Production. Development, testing, and pilot environments must be on separate network segments with firewall restrictions preventing access to production systems [18].

R4: Deploy EDR with After-Hours Monitoring. Financial institutions must deploy full EDR coverage and maintain 24/7 security monitoring capabilities [20].

R5: Implement React2Shell Detection Rules. IDS/IPS signatures must detect the following IoCs:

- Header: `Next-Action: `

- Body patterns: `$@0`, `$@1` (chunk references)

- Body patterns: `resolved_model`, `constructor:constructor`

- Body patterns: `_formData.get`, `process.mainModule`, `child_process.execSync` [11]

Regulatory Recommendations

R6: Mandate Zero-Trust Inter-Bank Security. The Central Bank of Nigeria must require mutual TLS with short-lived certificates for all inter-bank API calls, regardless of source.

R7: Establish Security Posture Score for Payment Rail Participants. Define minimum mandatory controls (patch critical CVEs within 14 days, no hardcoded secrets, network segmentation). Institutions falling below threshold must be suspended from NIBSS rails.

R8: Enforce Breach Notification Deadlines. The Nigeria Data Protection Commission must issue public enforcement actions for NDPA Section 38 violations. Fines of up to 2% of annual gross revenue must be applied [21].

Conclusion

On March 18, 2026, a single unpatched server at Sterling Bank answered a connection it should have refused. That connection cascaded through Nigeria's interconnected financial infrastructure, exposing 900,000 customer records, 657,242 identity documents, and cryptographic keys for 46 banks. The actor's own words capture the causal chain: "All access was permitted through Sterling Bank. Were it not for the compromise of Sterling, Remita would never have been hit" [1].

This paper has demonstrated that cascading breaches propagate not only through third-party vendors or software updates but also through peer trust corridors between domestic financial institutions. The security of interconnected financial infrastructure is only as strong as its weakest participant, and implicit trust relationships amplify rather than contain breaches.

Without zero-trust inter-bank architectures, mandatory secrets management, enforced breach notification, and independent verification of security controls, the next cascading breach is not a matter of if, but when. The 657,242 Nigerians whose identity documents are now in criminal hands cannot wait for the next breach to prompt action.

REFERENCES

1. Odes D. ByteToBreach exclusive Q&A: Why Sterling Bank, Remita, and CAC were targeted. Security Intelligence. 2026 Apr 16. Available from: https://bizwatchnigeria.ng/explainer-bytetobreach-hacker-sheds-llight-on-why-cac-sterling-bank-and-remita-were-targeted/
2. Odes D. Sterling Bank & Remita: How a global hacker walked through Nigeria's banking sector and took everything. Security Intelligence. 2026 Apr 8. Available from: https://securityintelligence.substack.com/p/sterling-bank-and-remita-how-a-global-f9c
3. Singh A, Yadav KS, Patel R, Sharma N. Internet-scale measurement of React2Shell exploitation using an active network telescope. arXiv:2603.12300. 2026 Mar 12.
4. Gkoulalas-Divanis A, Loukides G, Sun J. Publishing data from electronic health records while preserving privacy: A survey of algorithms. J Biomed Inform. 2014;50:4-19.
5. Crosby S, Goldberg S. The SolarWinds compromise: A supply chain attack on the US government. Harv Natl Secur J. 2022;13:1-56.
6. Chen Y, Huang C. SWIFT network security: Lessons from the Bangladesh Bank heist. J Financ Crime. 2020;27(3):789-804.
7. CyberArk. Deconstructing supply chain attacks and infiltrating a biohacker's mind. Black Hat. 2021. Available from: https://blackhat.com/sponsor-posts/07022021-deconstructing-supply-chain-attacks-and-infiltrating-a-biohackers-mind.html
8. Technext. Sterling Bank, Remita and CAC data breaches: Why did the Nigerian institutions say nothing? CoinMarketCap. 2026 Apr 16.
9. Girnus P, Patel D, Walsh J, Silva L, Verma A. CVE-2025-55182: React2Shell analysis, proof-of-concept chaos, and in-the-wild exploitation. Trend Micro Research. 2025 Dec 10.
10. National Vulnerability Database. CVE-2025-55182 Detail. NIST. 2025.
11. Aliyun Developer Community. React2Shell vulnerability automated credential theft attack mechanism and defense research. 2026 Apr 8.
12. Casey E. Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet. 3rd ed. Academic Press; 2011.
13. Lallie HS, Shepherd LA, Nurse JRC, et al. Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic. Comput Secur. 2021;105:102248.
14. Nigeria Data Protection Commission. Press Release on Investigation of Alleged Data Breach Involving Remita Payment Services Ltd., Sterling Bank and Other Entities. Abuja: NDPC; 2026 Apr 5.
15. MITRE. MITRE ATT&CK Framework, version 16. 2025.

ISSN No. 2454-6194 | DOI: 10.51584/IJRIAS |Volume XI Issue IV April 2026

16. Strom BE, Applebaum A, Miller DP, et al. MITRE ATT&CK: Design and philosophy. MITRE Corporation; 2018.
17. Sina Finance. From airport paralysis to financial penetration: Why supply chain attacks repeatedly succeed. 2025 Sep 23.
18. National Institute of Standards and Technology. Security and privacy controls for information systems and organizations. NIST SP 800-53, Rev. 5. NIST; 2018.
19. Okon EI, Uzoka FME. KYC data protection in Nigerian financial inclusion: Centralization risks and the attractive target asymmetry problem. J Bank Financ Technol. 2025;9(1):45-62.
20. Suleiman AT, Adekunle SO, Bello OR. Security posture and breach detection in Nigerian deposit money banks: An empirical survey of EDR deployment and SOC capabilities. Niger J Comput Sci. 2024;12(2):87-104.
21. Nigeria Data Protection Act, 2023. Federal Republic of Nigeria Official Gazette; 2023.