

AFRICAN RESILIENCE & DEVELOPMENT AGENCY (AFRDA)

AFRICA'S NEXT CYBER BATTLEFIELD

ZERO-CLICK ATTACKS SUPERCHARGED BY AI AGENTS

AUTHOR	AFRDA
ORGANIZATION	African Resilience & Development Agency (AFRDA)
PUBLICATION DATE	January 1, 2026

Contents

FOREWORD i

EXECUTIVE SUMMARY iii

1. Introduction 1

2. Documented Evolution of Zero Click Exploitation 4

3. AI Agents and Attack Amplification 7

4. EchoLeak and the AI Mediated Zero Click Chain 10

5. Organisational Conditions That Magnify Exposure 12

6. A Layered Defence Framework 14

7. African Policy and Institutional Priorities 19

8. Risk Trajectory and Future Threat Evolution 21

9. Conclusions and Recommendations 22

FOREWORD

Africa's digital transformation is extending communications, financial services, public administration and commercial activity across a rapidly expanding technology base. This expansion brings valuable opportunities for economic participation and institutional improvement. It also increases the number of devices, applications, data stores and automated processes that attract hostile attention. Cybersecurity therefore forms part of the foundation of digital development. Trust in digital services depends on the capacity of public institutions, businesses, technology providers and users to manage threats whose methods evolve alongside the systems they target.

Zero-click exploitation deserves particular attention because it changes the role of human behaviour in a cyber incident. Many established awareness programmes concentrate on suspicious links, unsafe attachments and deceptive requests. Zero-click attacks use automatic software processing to reach a device or service while the targeted person remains passive. The arrival of autonomous artificial intelligence agents extends this concern. An agent may read messages, retrieve files, summarise records, call connected tools and initiate actions at machine speed. A hostile instruction entering that operating context can turn legitimate capability into a channel for data exposure or harmful execution.

This report examines the convergence between zero-click techniques and AI agents. It traces documented incidents, explains the EchoLeak case involving Microsoft 365 Copilot, identifies the organisational conditions that magnify exposure and sets out a layered response. Its central finding concerns institutional design. Security depends on the permissions granted to agents, the boundaries around data and tools, the quality of monitoring, the speed of vulnerability remediation and the authority assigned to security governance. Human vigilance retains value across many threats and automated attack paths require technical and organisational controls that operate before a person enters the sequence.

The policy significance for Africa is immediate. Governments and enterprises are adopting cloud services, mobile platforms and AI-enabled applications during a period of expanding digital ambition and procurement decisions made during this period will shape security exposure for years. Effective action therefore requires early attention to architecture, identity, access, testing,

incident response and accountability. Shared learning among national cyber authorities, regulators, technology providers, research institutions and users can strengthen this foundation and support confident adoption of digital innovation. AFRDA presents this report as a contribution to that effort to help decision makers understand the threat mechanism and translate that understanding into practical safeguards.

Bakaluba B. Daniel
Continental Coordinator / CEO
African Resilience & Development Agency (AFRDA)

EXECUTIVE SUMMARY

Zero-click attacks exploit automatic processing inside software, messaging systems and connected services. The targeted person performs zero action in the attack chain. This feature changes the practical value of controls centred on user judgement. Awareness training can reduce phishing exposure. It has limited effect when a vulnerable application processes hostile content immediately after receipt. Documented incidents involving Android's Stagefright media framework, WhatsApp's calling stack and Apple's iMessage service demonstrate the reach of this technique across major platforms. Each incident used a different technical pathway. Their shared lesson lies in the transfer of control from the user to software that processes incoming content automatically.

Artificial intelligence agents deepen this exposure because they combine information access with delegated capability. An enterprise agent may search corporate records, read email, summarise documents, use external services and call business tools. These functions increase productivity by compressing tasks that previously required several human steps. They also create a pathway through which hostile content can influence automated reasoning and action. Indirect prompt injection represents one important mechanism. An attacker places instructions inside content that an agent later retrieves or processes. The agent may interpret those instructions as part of its task, gain access to information available in its operating context and produce an output that serves the attacker's objective.

The 2025 EchoLeak research on Microsoft 365 Copilot illustrates this mechanism. Researchers constructed a chain in which a crafted email entered an enterprise environment and influenced Copilot through indirect prompt injection. The chain used the agent's access to organisational information and a sequence of control bypasses to support data exfiltration. Microsoft assigned CVE-2025-32711 to the vulnerability and deployed a service-side fix before public disclosure. The case remains important because it demonstrates a general security problem: an AI system can treat externally supplied content as instructions and then use trusted permissions in a hostile sequence. The attacker's leverage comes from the agent's position inside the organisation.

Governance conditions increase the significance of this technical risk. IBM's 2025 Cost of a Data Breach research reported that 63 per cent of the 600 organisations in its study lacked an AI governance policy or were developing

one. Among organisations that experienced an AI-related security incident, 97 per cent reported gaps in AI access controls. These findings describe a period in which deployment has often advanced ahead of formal oversight. The resulting exposure includes unknown AI use, excessive permissions, weak inventories, limited logging, fragmented responsibility and incomplete incident plans.

The analysis identifies seven connected control priorities. Organisations should maintain an authoritative inventory of AI systems and the data, tools, identities and external services connected to each system. They should apply least privilege to every agent identity and separate routine tasks from privileged actions. High-impact actions should require explicit human approval. Security architecture should distinguish trusted instructions from external content, validate input and output, apply data loss prevention controls and record every consequential agent action. Patch management should give urgent treatment to remotely exploitable vulnerabilities and vendor advisories. Adversarial testing should examine indirect prompt injection, excessive agency, data exfiltration routes and failures across tool boundaries. Governance should assign clear authority to executive leadership, security teams, system owners, data stewards, procurement units and incident responders.

African public institutions and enterprises can strengthen collective resilience through procurement standards, coordinated vulnerability disclosure, national guidance for AI security, sector exercises and regional exchange among cyber authorities. Organisations with limited specialist capacity can begin with a focused baseline: identify deployed agents, reduce permissions, protect sensitive repositories, require approval for consequential actions, preserve logs and establish an emergency process for disabling risky integrations. These measures address the mechanism through which AI amplifies zero-click exposure.

The risk trajectory points towards attacks with greater speed, broader reach and deeper access to organisational processes. Security policy therefore requires continuous adaptation. The objective is controlled agency: AI systems should perform defined tasks within verified boundaries, under accountable ownership and with evidence available for oversight. This model supports digital innovation through a stronger institutional foundation.

ABBREVIATIONS

Abbreviation	Meaning
AI	Artificial intelligence
CERT	Computer Emergency Response Team
CVE	Common Vulnerabilities and Exposures
DLP	Data loss prevention
LLM	Large language model
MMS	Multimedia Messaging Service
RAG	Retrieval augmented generation
VoIP	Voice over Internet Protocol

1. Introduction

1.1. The emerging security problem

Cybersecurity practice has long devoted substantial attention to actions taken by users. Phishing simulations, awareness campaigns and email warnings aim to help people recognise hostile links, attachments and requests. These measures remain useful across many attack categories. Zero-click exploitation follows a different route. The attacker targets software that automatically receives, parses, displays, indexes, summarises or otherwise processes content. The vulnerable system performs the triggering action. The person associated with the device or account remains passive throughout the critical stage.

This change matters because it shifts the centre of defence towards architecture and institutional control. Secure coding, rapid patching, isolation, access management, content handling and monitoring become decisive. An employee may follow every awareness instruction and still face compromise through a vulnerable media library, messaging component or automated assistant. The organisation therefore carries direct responsibility for selecting systems, configuring permissions, evaluating suppliers and maintaining a response capability suited to automated attack paths.

The expansion of autonomous AI agents adds a second layer to this problem. These systems receive natural-language instructions, gather context, interpret information and use connected capabilities to complete tasks. Their value grows with access. An agent becomes more useful when it can reach email, documents, calendars, databases and operational tools. The same access increases the consequences of manipulation. A hostile input can travel through the agent's trusted position and reach information or functions available to that agent.

This report describes the resulting amplification effect. Zero-click techniques provide a route into automatic processing. AI agents provide scale, context and delegated action. Their convergence creates attack chains that can move from message receipt to information retrieval and exfiltration within an automated workflow. The organisational challenge concerns the design of boundaries around that workflow.

1.2. Definition and analytical scope

A zero-click attack is an attack whose critical execution path requires zero action from the targeted user. The attacker initiates the sequence, and an application or service processes the hostile input automatically. The term covers several technical mechanisms. A memory-corruption flaw may allow remote code execution. A parser may mishandle a crafted file. An AI system may ingest an indirect instruction from email, a webpage, a document, an image or retrieved data. These mechanisms differ at the code level. They share an operational characteristic: software activity replaces a user decision as the trigger.

An AI agent is a software system that uses an AI model to interpret a goal, gather information and perform actions through connected tools or services. Agency exists along a spectrum. Some systems prepare suggestions for human approval. Others execute steps automatically within an assigned scope. Risk grows with four features: access to sensitive context, authority to call tools, freedom to select actions and connectivity to external content. An agent combining these features occupies a powerful institutional position.

The report focuses on enterprise and public-sector exposure created by the interaction of automatic content processing, AI reasoning and delegated permissions. It examines documented zero-click incidents as evidence of the underlying pattern. It then uses EchoLeak as a case study of AI-mediated exploitation. The final sections analyse organisational constraints and propose a layered response suited to African digital development. The report addresses defensive policy and system governance. Detailed exploit construction falls outside its purpose.

1.3. Why the issue matters for African digital transformation

African economies are expanding their use of mobile services, cloud platforms, digital payments, online public services and data-driven administration. AI functions are entering this environment through global software suites, customer-service systems, analytical tools and locally developed applications. Each deployment creates relationships among users, vendors, data repositories, networks and automated services. Security outcomes depend on the design of those relationships.

The consequences of compromise can extend beyond an individual device. An agent connected to financial records, citizen data, health information, research

files or operational systems may provide a path into high-value institutional assets. Resource constraints can heighten this exposure where organisations rely on small security teams, inherited software, delayed patching and limited visibility across third-party services. Procurement and deployment decisions therefore carry long-term importance.

The policy response requires attention at organisational, national and regional levels. Enterprises control identities, permissions, logging and incident preparation. Vendors control product architecture, vulnerability remediation and security documentation. Governments shape procurement rules, reporting expectations, critical-infrastructure requirements and national response arrangements. Research institutions and professional communities contribute testing, skills and public knowledge. A resilient system emerges when these responsibilities reinforce one another.

2. Documented Evolution of Zero Click Exploitation

2.1. Stagefright and automatic media processing

The Stagefright vulnerabilities disclosed in 2015 drew attention to the scale of exposure created by automatic media handling on Android devices. The affected media framework processed several formats used in multimedia messages and other content. A crafted media file could trigger memory corruption and, for critical flaws, support remote code execution in the media service. Estimates at the time placed hundreds of millions of Android devices within the affected population. The scale reflected the wide distribution of the vulnerable component and the fragmented process through which device updates reached users.

Stagefright established an important institutional lesson. A message delivery system can place hostile content directly before a parser that acts ahead of the user. The security boundary therefore sits inside the receiving application and the operating system. User caution enters the sequence after vulnerable processing has already occurred. Defence depends on code quality, component isolation, update availability and the speed with which manufacturers, network operators and users apply remedies.

The incident also showed how supply chains shape vulnerability management. Google could publish patches for the Android platform, while device manufacturers and carriers controlled distribution across many products. Delay at any stage prolonged exposure. This pattern remains relevant to AI-enabled systems, where model providers, application vendors, cloud operators, integration partners and customers share responsibility across a layered service.

2.2. Pegasus and the WhatsApp calling stack

In 2019, WhatsApp addressed CVE-2019-3568, a buffer overflow in its VoIP calling stack. Reporting connected exploitation of the flaw with Pegasus spyware. A specially prepared call could reach the vulnerable component and support code execution on the target device. The target's decision to answer the call carried little relevance to the exploit path. The calling stack handled the incoming request and created the opening.

Pegasus illustrates the high consequences associated with remote compromise of a mobile device. Spyware with deep access can reach

messages, stored information, sensors and account activity. The value of the exploit therefore came from the relationship between a narrow entry point and the extensive privileges available after successful installation. Security teams face a comparable concern with AI agents. A single hostile input can carry greater consequence when the receiving system holds broad permissions and access to sensitive context.

The incident also demonstrates the role of rapid vendor action and coordinated notification. WhatsApp released updated versions and urged users to apply them. Organisations responsible for mobile fleets require processes that identify affected versions, accelerate deployment and verify completion. A public advisory creates value through timely operational response.

2.3. FORCEDENTRY and iMessage

Citizen Lab's 2021 analysis of FORCEDENTRY documented a zero-click exploit delivered through iMessage and associated with Pegasus. The exploit used a maliciously crafted PDF presented to Apple's image-processing system. Citizen Lab supplied technical information to Apple, which assigned CVE-2021-30860 and released security updates. The incident showed the sophistication available to well-resourced operators and the continuing attraction of messaging services whose normal operation involves automatic content processing.

FORCEDENTRY adds a further lesson about defence in depth. Platform-level protections can raise the cost of exploitation. Attackers may still combine several weaknesses to cross those boundaries. Security architecture therefore gains strength from independent layers: safe parsing, process isolation, memory protections, restricted privileges, behavioural detection and rapid remediation. The failure of one layer then meets further controls before the attacker reaches valuable assets.

2.4. The cross platform lesson

Stagefright, the WhatsApp VoIP flaw and FORCEDENTRY arose in distinct products and technical components. Their significance extends beyond the named platforms. Automatic processing appears across messaging, document preview, media rendering, email security, cloud indexing and AI-assisted summarisation. Each function accepts content and performs work before a person makes a security decision. This recurring design pattern creates a broad class of exposure.

The incidents also reveal a relationship between privileges and impact. Entry through a parser becomes severe when the compromised process can reach powerful functions or sensitive data. Least privilege and isolation therefore limit the consequences of successful exploitation. Patching addresses the specific flaw. Architectural restraint reduces the value of the next flaw to an attacker.

For African institutions, the practical message concerns asset management and update governance. A security team requires visibility into devices, operating systems, applications, versions and vendor support. It also requires authority to deploy urgent fixes across operational units. Fragmented ownership, legacy equipment and unclear maintenance contracts can delay this work. Procurement agreements should specify support periods, vulnerability notification, patch delivery and assistance during major incidents.

3. AI Agents and Attack Amplification

3.1. Productivity through delegated capability

AI agents can reduce the time required for information-intensive work. They can search large collections, prepare summaries, classify requests, draft responses, extract fields and coordinate steps across software tools. In public administration, similar capabilities may support case handling, document review and citizen services. In business, they may assist sales, finance, procurement, customer support and technical operations. The productivity gain arises from the combination of model reasoning, access to context and permission to act.

The same combination defines the security exposure. Context may include confidential email, internal files, customer records and credentials embedded in operational workflows. Tool access may include messaging, database queries, code execution, payments or changes to system state. Autonomy allows the agent to select a sequence of steps at machine speed. A manipulated agent can therefore magnify the reach of a hostile instruction beyond the content in which that instruction arrived.

This amplification differs from a conventional software vulnerability in one important respect. Natural-language content can influence the agent's behaviour because the system uses language as data and instruction. A message intended for summarisation may contain text that attempts to redefine the task. A webpage retrieved for research may direct the agent towards an external destination. A document may contain concealed text designed for machine processing. The application must preserve the distinction between trusted instructions and external material across every stage of the workflow.

3.2. Indirect prompt injection

Indirect prompt injection occurs when hostile instructions enter an AI system through external content. The attacker may place the content in email, a document, a website, an image, a shared record or data returned by a connected tool. The agent retrieves or processes the material during a legitimate task. The model may then treat the hostile instruction as relevant guidance and alter its behaviour.

The technique gains strength from ambiguity in language-model processing. The model receives tokens representing system instructions, user requests, retrieved content and tool output. Security depends on the surrounding application preserving provenance and enforcing policy outside the model. Prompt wording alone provides a fragile boundary. Deterministic controls must decide which tools the agent may call, which records it may retrieve, which destinations it may contact and which outputs require review.

The impact of prompt injection therefore depends on system design. An agent limited to drafting text inside an isolated workspace presents a narrower risk. An agent with broad data access, external communication and permission to execute actions presents a larger risk. The vulnerability may begin in language interpretation. Its consequences emerge through identity, access, application integration and data-flow architecture.

3.3. The governance gap

Rapid AI adoption has often preceded the creation of formal governance. IBM's 2025 research, based on breaches experienced by 600 organisations between March 2024 and February 2025, reported that 63 per cent lacked an AI governance policy or were developing one. The same research found access-control gaps among 97 per cent of organisations that experienced an AI-related security incident. These figures describe institutions already managing breach conditions. They therefore offer evidence about the relationship between weak oversight and exposure among affected organisations.

An AI governance policy provides value when it changes operational practice. Useful governance creates an inventory of systems, assigns owners, classifies use cases, defines approval requirements, sets data-access rules, establishes testing standards and links AI incidents to the wider cyber response process. A policy document with limited implementation leaves the underlying exposure in place. Executive sponsorship and measurable controls give the framework practical force.

Shadow AI adds a further challenge. Staff may adopt external tools through personal accounts, browser extensions or unsanctioned applications. Security teams then lose visibility into data transfers, retention terms, model behaviour and connected services. Organisations need approved alternatives, clear rules and monitoring proportionate to risk. A punitive approach may drive use further from view. Accessible secure tools and credible oversight can bring activity into managed channels.

3.4. Institutional actors and responsibility

Executive leadership sets the organisation's risk appetite and determines the resources available for security. The board or governing body requires evidence about deployed AI systems, critical dependencies, major incidents and remediation progress. Security leadership translates this oversight into architecture, monitoring and response. Data stewards control the classification and use of sensitive information. System owners manage business purpose, configuration and performance. Procurement teams shape vendor obligations before deployment. Legal and compliance functions interpret regulatory duties. Internal audit tests the operation of controls.

Technology providers carry responsibility for secure design, vulnerability handling, transparent documentation and timely remediation. Their service architecture may remain largely invisible to customers, which increases the importance of contractual assurance and independent testing. National cyber authorities and sector regulators can support common guidance, incident exchange and minimum expectations for critical services. Universities, research centres and professional associations can strengthen the supply of specialist skills and contribute locally relevant testing.

Effective governance connects these actors through decisions and evidence. A deployment should have a named owner, an approved purpose, a data classification, an access model, test results, a monitoring plan and an exit procedure. Incident responders should know how to isolate the agent, revoke its credentials, preserve evidence and communicate with the vendor. This clarity reduces delay during a fast-moving event.

4. EchoLeak and the AI Mediated Zero Click Chain

4.1. The case

EchoLeak was the name given to CVE-2025-32711, a vulnerability chain affecting Microsoft 365 Copilot. Security researchers at Aim Security reported the issue to Microsoft in January 2025. Microsoft deployed a service-side remedy before public disclosure in June 2025. The research presented a proof of concept in which a crafted email could influence Copilot through indirect prompt injection and support the exfiltration of information available in the Copilot context.

The case joined several weaknesses across the application. The malicious content entered through email. Copilot could retrieve that content during normal operation. The injected instructions sought sensitive information from the model's accessible context. Additional techniques attempted to bypass prompt-injection detection, link protections and content-security restrictions. The chain used trusted Microsoft services to support an outbound path. Its effectiveness depended on several components behaving together.

EchoLeak therefore offers a stronger analytical lesson than a single product flaw. It shows how an AI assistant can become the connecting layer among untrusted content, internal data and an external destination. The agent's access supplies the context. The model's interpretation supplies the behavioural shift. Application integrations supply the path. A secure design must control each relationship.

4.2. Attack sequence

The sequence begins when an attacker sends a crafted email into an organisation. The visible message can resemble ordinary correspondence. Hidden or strategically placed instructions target the AI system that may later process the content. Concealment can use formatting, markup, metadata or other presentation techniques that preserve machine-readable text while reducing human visibility.

The AI assistant encounters the message during retrieval, summarisation or another task. The hostile instruction attempts to change the purpose of the agent's work. It may request information from earlier correspondence, connected files or other sources available to the user's Copilot context. The attack then needs a route that carries the resulting information towards

infrastructure controlled by the attacker. EchoLeak research demonstrated a chain of bypasses supporting that route.

The user plays a limited role in this sequence. The critical events arise from message delivery, automated retrieval and agent processing. This feature supports the zero-click description. The proof of concept also illustrates a scope violation. The agent begins with an authorised enterprise function and moves into conduct serving an external attacker. The transition occurs through content that the system receives as data.

4.3. Security implications

The first implication concerns inherited access. Microsoft 365 Copilot operates within the permissions available to the user. Broad or outdated user access therefore expands the information that the assistant may retrieve. Identity governance for human accounts directly affects AI exposure. Reviews of group membership, shared repositories and legacy permissions become part of agent security.

The second implication concerns automatic trust between components. An email system, retrieval layer, model, rendering engine and external service may each apply separate controls. Attackers search for gaps between them. A link filter may inspect one representation while another component resolves a different representation. A content-security rule may allow a trusted service that can be repurposed as a relay. Security testing must examine the full sequence across component boundaries.

The third implication concerns detection. A conventional alert may search for malware, unusual login activity or a known network signature. Prompt injection can appear as natural language inside legitimate content. The agent may use valid credentials and approved services. Detection therefore requires telemetry about agent inputs, retrieved sources, tool calls, sensitive-data access and outbound destinations. Correlation across these events can reveal behaviour that each component regards as ordinary.

The fourth implication concerns generalisability. Email served as the delivery channel in EchoLeak. Comparable conditions can arise in websites, shared documents, collaboration platforms, ticketing systems, code repositories and multimodal content. Any source that an agent reads can carry instructions. Any tool that an agent controls can extend the consequence. Security policy should therefore address the class of mechanism across AI applications.

5. Organisational Conditions That Magnify Exposure

5.1. Excessive permissions

AI systems often inherit access from users, service accounts or connected applications. Historical permissions can accumulate as staff change roles, teams share repositories and projects end. An agent operating through this access can retrieve information far beyond the needs of its current task. Excess privilege expands the potential result of prompt injection, account compromise and accidental disclosure.

Permission management should begin with a clear task definition. Each agent needs a dedicated identity and an access package matched to that task. Privileged operations should use separate credentials with short validity and explicit approval. Regular access reviews should compare actual use with assigned rights. Dormant integrations and legacy tokens should move promptly into revocation.

5.2. Weak separation of instructions and data

LLM applications frequently combine system prompts, user requests, retrieved documents and tool responses within a shared processing context. This design supports flexible reasoning. It also creates ambiguity about authority. Hostile text inside retrieved content may compete with the instruction governing the task.

Architecture should preserve provenance from ingestion through output. External content needs clear labels and isolation. Policy engines outside the model should enforce tool permissions and data rules. Structured schemas can constrain inputs and outputs. Sensitive operations should depend on deterministic validation. These controls place authority in application logic and reduce reliance on the model's interpretation of conflicting language.

5.3. Limited visibility and fragmented ownership

Many organisations lack a complete inventory of AI tools, agents, plugins, model endpoints and data connections. Business units may deploy services through ordinary software subscriptions. Staff may activate AI features inside platforms already approved for another purpose. Security teams then face an incomplete picture of the operating environment.

Fragmented ownership compounds the problem. A business unit may own the use case, information technology may manage the platform, a cloud provider

may operate the model and a separate vendor may supply an integration. Each actor sees part of the risk. A central register and a defined approval process can connect these responsibilities. The register should identify the system owner, model provider, data sources, tools, identities, deployment environment, risk rating, testing status and emergency shutdown procedure.

5.4. Delayed patching and dependency risk

Zero-click vulnerabilities often create urgency because they expose a service during ordinary operation. Patch delay extends the period available for exploitation. Organisations may face compatibility testing, distributed devices, vendor dependencies and operational constraints. These practical conditions require a prioritisation process suited to exploitability and potential impact.

AI applications add dependencies across cloud services, model APIs, libraries, plugins and connectors. A service-side remedy may arrive directly from a provider. A customer-managed component may require local action. Security teams need reliable advisory channels and ownership for each dependency. Contracts should define notification, remediation support and evidence of closure. Asset inventories should connect affected products with responsible teams.

5.5. Gaps in testing and incident preparation

Functional testing confirms that an agent completes its intended task. Security testing examines how the system behaves under hostile input, ambiguous instructions, excessive data access, tool misuse and failures across integrations. Many deployment processes give greater attention to functionality than adversarial behaviour. This imbalance leaves attack paths undiscovered until research or an incident exposes them.

Incident plans also require adaptation. Responders need procedures for suspending agent activity, revoking machine credentials, isolating connectors, preserving prompts and tool logs, identifying exposed records and coordinating with service providers. Exercises should include scenarios in which approved accounts and trusted services carry hostile activity. The evidence trail must support technical investigation, regulatory assessment and communication with affected parties.

6. A Layered Defence Framework

6.1. Governance and system inventory

Every organisation using AI agents should maintain a current register of deployments. The register should cover approved systems and observed shadow use. It should map each agent to its owner, business purpose, model, deployment environment, data sources, permissions, connected tools, external destinations and vendor contacts. Risk classification should reflect the sensitivity of accessible information and the consequence of available actions.

A governance committee or designated authority should approve high-risk use cases and review material changes. The review process should require a security assessment, privacy assessment, data classification, test evidence, monitoring design and incident procedure. Renewal at defined intervals can capture changes in models, integrations and business use. Executive reporting should present deployment numbers, unresolved high-risk findings, permission-review status, incidents and remediation progress.

6.2. Capability limitation and isolation

Agent capability should match the minimum functions required for the approved task. Sandboxing can restrict file access, network destinations, code execution and interaction with operating-system resources. Separate environments can protect sensitive systems from experimental agents. Tool allowlists can confine the agent to approved functions. Rate limits and transaction limits can reduce the scale of harmful activity.

High-impact capabilities deserve additional separation. An agent that drafts a payment instruction should use a distinct process from the system that authorises payment. An agent that analyses records should have read access suited to the assignment and a controlled route for exporting results. Separation reduces the ability of one manipulated component to complete an entire harmful sequence.

6.3. Identity and access control for agents

Each agent should operate through a distinct machine identity. Shared service accounts weaken accountability because logs then provide limited evidence about the actor behind an event. Dedicated identities support precise permissions, credential rotation, monitoring and revocation. Short-lived tokens reduce the period of exposure after compromise.

Access reviews should include the agent, the sponsoring user and the underlying data repositories. Role changes should trigger updates across these layers. Sensitive information should remain available to a narrow set of identities. Attribute-based policies can add conditions involving location, device posture, time, data classification and transaction value. Privileged access management can protect administrative functions and high-risk tools.

6.4. Human approval for consequential actions

Human approval should govern actions that carry significant financial, legal, operational or privacy consequences. Examples include payments, deletion of records, changes to user privileges, disclosure of sensitive data, publication of official decisions and control of physical systems. The approval interface should present the proposed action, the source material, the data involved and the destination. This information helps the reviewer make an informed decision.

Approval quality depends on timing and clarity. A frequent stream of low-value requests can create fatigue. Risk-based thresholds should concentrate attention on consequential actions. The system should preserve evidence of the decision, the identity of the reviewer and the final execution. Emergency processes should support rapid suspension when abnormal activity appears.

6.5. Input handling and content provenance

External content should enter the agent environment with an explicit provenance label. The application should preserve the source across retrieval, transformation and output. Trusted instructions should occupy a protected channel controlled by the system owner. Email, websites, documents and tool responses should remain data within the workflow.

Input scanning can identify suspicious markup, concealed text, unexpected encoding, hostile URLs and known prompt-injection patterns. Scanning provides one layer. Structural controls provide broader protection because attackers can vary wording and presentation. The application should constrain the influence of retrieved content and route unusual requests for review. Multimodal systems require inspection of text, images, audio and embedded objects.

6.6. Output validation and data loss prevention

Outputs require validation before they reach external users, systems or network destinations. Data loss prevention controls can detect credentials,

financial records, personal data and classified information. Structured output schemas can limit fields and formats. Destination controls can restrict outbound links, image retrieval, web requests and message delivery. Sensitive content can move into a quarantine queue for authorised review.

The control should also compare the output with the purpose of the task. A summarisation agent attempting to transmit a credential presents a clear scope breach. An agent reaching an unfamiliar domain during an internal research task deserves investigation. Context-aware policies can identify these relationships and reduce dependence on simple keyword matching.

6.7. AI gateway and firewall controls

An AI security gateway can centralise policy enforcement across model requests and responses. It can authenticate applications, apply rate limits, inspect content, enforce model selection, record prompts, filter sensitive information and restrict external destinations. This function resembles a specialised control plane for AI interactions.

Its effectiveness depends on architecture and coverage. Direct connections that bypass the gateway create gaps. Encrypted traffic, multimodal content and tool calls require appropriate inspection points. The gateway also becomes a critical component that needs resilience, access protection and monitoring. Organisations should treat it as one layer within a wider control system.

6.8. Patch and vulnerability management

Patch management should classify zero-click and remotely exploitable vulnerabilities as urgent. Security teams should subscribe to vendor advisories, connect assets to software versions and define accelerated testing for critical fixes. Deployment records should show coverage and exceptions. Executive escalation should address systems that remain exposed after the target period.

Vulnerability management should also cover AI-specific components. Model gateways, orchestration frameworks, plugins, vector databases, retrieval pipelines and browser automation can each introduce exploitable flaws. Software bills of materials and dependency scanning can improve visibility. Coordinated disclosure channels give researchers a safe route for reporting findings.

6.9. Monitoring and behavioural detection

Effective monitoring records the agent's identity, originating request, retrieved sources, data accessed, tool calls, policy decisions, outputs and network destinations. Logs should support correlation across these events. Sensitive prompt content may require protection and retention limits. Security teams need enough detail to reconstruct consequential behaviour.

Behavioural analytics can identify unusual access volume, new destinations, repeated policy violations, unexpected tool sequences and activity outside normal operating periods. Baselines should reflect the purpose of each agent. A research assistant and a payment-processing agent will produce different patterns. Alerts gain value when they connect behaviour with data sensitivity and action consequence.

6.10. Adversarial testing and exercises

Security testing should begin before deployment and continue after material changes. Red teams should examine indirect prompt injection, hidden instructions, tool abuse, data exfiltration, identity escalation, cross-tenant exposure and failures in content provenance. Tests should cover the full application chain and the relationships among components.

Tabletop exercises should bring together system owners, security teams, legal advisers, communications staff, executives and vendor contacts. A useful scenario can begin with a crafted message, proceed through abnormal agent activity and develop into suspected data exposure. Participants should practise containment, evidence preservation, impact assessment, notification and service restoration. Lessons should feed into architecture and policy.

6.11. Zero trust implementation

Zero trust provides a useful organising principle for AI security. Every request, identity, device, content source and tool call should undergo verification appropriate to its risk. Access should reflect current context and minimum task requirements. Continuous evaluation should replace permanent assumptions about trust.

For AI agents, this means verifying the agent identity, the source of instructions, the sensitivity of requested data, the legitimacy of the tool call and the destination of any output. Policy engines should make these decisions through

observable rules. The model can support analysis. Authorisation remains an institutional control.

7. African Policy and Institutional Priorities

7.1. Public procurement and critical services

Public procurement can establish security requirements before AI systems enter government operations. Tender documents and contracts should require disclosure of data flows, deployment locations, subcontractors, model providers, retention practices, security testing and vulnerability-response arrangements. Buyers should request evidence of identity controls, logging, incident support and service termination procedures. Critical services require stronger assurance and independent testing.

Procurement teams need technical support to evaluate these claims. National digital agencies, cyber authorities and sector regulators can develop common clauses and assessment guidance. Shared standards reduce duplication across ministries and improve the negotiating position of smaller institutions. Contract management should verify continuing compliance throughout the service period.

7.2. National guidance and incident coordination

National cybersecurity strategies should address AI agents as part of the evolving attack surface. Guidance can define minimum controls for government entities and critical infrastructure. National CERTs can incorporate AI-related incidents into reporting channels, advisories and exercises. Sector-specific regulators can translate general principles into requirements suited to finance, health, energy, telecommunications and public administration.

Cross-border exchange has practical value because vendors and attack methods operate across jurisdictions. Regional bodies and professional networks can share indicators, lessons from incidents, tested controls and training resources. A common vocabulary for AI incidents can improve the quality of reporting and comparison.

7.3. Skills and research capacity

AI security requires expertise spanning cybersecurity, software engineering, data governance, model behaviour, law and organisational risk. Universities and training institutions can integrate these fields into curricula and professional education. Applied research should examine local deployments, African languages, regional service providers and sector conditions. This work can identify risks that global product testing may overlook.

Responsible vulnerability research also needs supportive institutions. Clear disclosure policies, safe reporting channels and recognition for researchers can improve early detection. Partnerships among universities, CERTs, vendors and public agencies can turn findings into remediation and wider guidance.

7.4. A practical baseline for resource constrained organisations

Organisations with limited security capacity can begin with a concentrated set of controls. They should identify every approved AI tool and connected data source, assign a named owner, restrict permissions, protect sensitive repositories, require human approval for consequential actions and preserve logs. They should also establish a contact route for vendor support and a procedure for disabling an agent or connector during an incident.

This baseline creates visibility and containment. Additional maturity can follow through formal risk classification, automated monitoring, adversarial testing, DLP integration and independent assurance.

8. Risk Trajectory and Future Threat Evolution

AI agents are moving towards deeper integration with enterprise data and operational tools. Greater autonomy can shorten task completion and expand the range of functions handled by software. It can also compress the time between hostile input and harmful action. An attack sequence that once required manual reconnaissance, data collection and transmission may unfold through automated steps inside a trusted account.

Attackers are likely to refine concealment, adapt prompts across languages and modalities, and search for inconsistencies among filters, models, connectors and rendering systems. They can use AI to generate variations and test responses at scale. Defenders can also use AI for code analysis, anomaly detection, triage and response. The institutional advantage will depend on governance, data quality, visibility and the speed with which lessons become controls.

The growth of agent-to-agent communication introduces further complexity. One agent may rely on information or actions supplied by another. A hostile instruction can then travel across a chain of systems. Identity, provenance and authorisation must remain intact across these exchanges. Organisations will need records showing which agent initiated an action, which source influenced it and which policy approved it.

Vulnerability concentration may also increase where many institutions depend on the same cloud services, model providers or orchestration frameworks. A weakness in a widely used component can create broad exposure. Diversification, contingency planning and transparent vendor risk management can reduce systemic impact. National authorities should consider these dependencies within critical-infrastructure planning.

The threat will therefore persist as a governance issue alongside its technical form. Models and products will change. The core questions remain stable: which content can influence the system, which data can the system reach, which actions can it take, which destinations can it contact, who owns the risk and which evidence supports oversight. Institutions that answer these questions clearly will enter future deployments with stronger control.

9. Conclusions and Recommendations

9.1 Conclusions

Zero-click attacks shift the decisive security event from user behaviour to automatic processing. Stagefright, the WhatsApp VoIP vulnerability and FORCEDENTRY show how hostile content can exploit widely used communications systems during routine operation. These incidents establish the importance of secure parsing, isolation, rapid patching and restricted privileges.

AI agents increase the consequence of this pattern by connecting external content with sensitive context and delegated tools. Indirect prompt injection gives attackers a route for influencing an agent through material retrieved during a legitimate task. EchoLeak demonstrated how a crafted email could participate in a wider chain involving Microsoft 365 Copilot, organisational data and an outbound exfiltration route. The case establishes a general design concern extending across agentic applications.

Organisational exposure grows through excessive permissions, weak provenance, fragmented ownership, limited monitoring, delayed remediation and gaps in adversarial testing. Governance policy has practical value when it produces named responsibility, an accurate inventory, enforceable access rules, test evidence and incident readiness. The IBM findings on AI governance and access controls indicate a substantial implementation gap among surveyed organisations with breach experience.

African institutions have an opportunity to address these conditions during an important period of digital expansion. Procurement standards, national guidance, regional exchange, skills development and local research can strengthen the environment in which organisations deploy AI. The objective is a form of controlled agency in which automated systems act within defined authority and produce evidence suitable for oversight.

9.2. Recommendations

9.2.1 Establish an authoritative AI system register

Every public institution and enterprise should maintain a register covering agents, models, plugins, connectors, data sources and external destinations. Each entry should identify an accountable owner, approved purpose, risk

classification, permissions, vendor contact and emergency shutdown process. The register should support periodic review and executive reporting.

9.2.2 Enforce least privilege for human and machine identities

System owners should match permissions to the approved task and review inherited user access that influences agent retrieval. Each agent should use a distinct identity with short-lived credentials where feasible. Privileged functions should use separate authorisation and stronger monitoring.

9.2.3 Require approval for consequential actions

Organisations should place human approval before payments, deletion, privilege changes, external disclosure, official publication and control of physical systems. Approval interfaces should display the source, proposed action, affected data and destination. Decision records should support audit and investigation.

9.2.4 Protect the boundary between instructions and external content

Application designers should preserve content provenance and isolate trusted system instructions from email, websites, documents and tool output. Deterministic policy engines should control data access, tool use and outbound connections. Structured inputs and outputs should limit ambiguity.

9.2.5 Inspect input output and data movement

Security teams should deploy input scanning, output validation, DLP controls and destination restrictions around AI applications. Monitoring should connect requests, retrieved content, data access, tool calls and network activity. High-risk anomalies should trigger containment and review.

9.2.6 Accelerate remediation of remotely exploitable vulnerabilities

Patch programmes should give urgent priority to zero-click and remote-code-execution flaws. Asset inventories should connect affected versions with responsible owners. Procurement contracts should specify vulnerability notification, fix delivery, customer support and evidence of closure.

9.2.7 Conduct continuous adversarial testing

Organisations should test prompt injection, hidden content, excessive agency, tool misuse, data exfiltration and failures across service boundaries. Testing

should occur before deployment and after material changes. Findings should enter tracked remediation with accountable owners.

9.2.8 Integrate AI incidents into response plans

Incident procedures should cover suspension of agents, credential revocation, connector isolation, log preservation, data-impact assessment and vendor coordination. Exercises should involve technical teams, business owners, legal advisers, communications staff and executive decision makers.

9.2.9 Strengthen national and regional coordination

Governments should issue AI security guidance for public bodies and critical services. National CERTs should include AI incidents in reporting, advisories and exercises. Regional networks should support exchange of technical lessons, indicators, training material and tested procurement clauses.

9.2.10 Invest in African AI security capacity

Universities, research organisations, professional bodies and public agencies should develop training and applied research linking cybersecurity, data governance, AI systems and institutional risk. Support for coordinated vulnerability disclosure can connect local research with timely remediation.

For More Information

For additional information about this thechnical paper or to engage with AFRDA, please contact:

African Resilience & Development Agency (AFRDA)

info@afrda.org

www.afrda.org

Document Classification: Public

Last Updated: January 2026

Version: 1.0