IJARIIT International Journal Of Advance Research, Ideas And Innovations In Technology

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 11, Issue 5 - V11I5-1190)
Available online at: https://www.ijariit.com

Fortifying AI Infrastructure: Securing Code, Configuration, and Integrity in National Systems

Ifeoma Eleweke
i.eleweke.211@westcliff.edu
Westcliff University, California

ABSTRACT

The rapid adoption of artificial intelligence (AI) on cloud platforms, such as AWS and Azure, has introduced critical security vulnerabilities across various national sectors, including defense, healthcare, and energy. While these environments deliver scalable intelligence, they also expand the attack surface, exposing misconfigured resources, unverified code, and weak identity controls. Recent breaches, including Capital One's AWS data exposure, Tesla's compromised Kubernetes console, and Microsoft's AI dataset leak, demonstrate how cloud-hosted AI pipelines can be weaponized through insecure defaults, leaked credentials, and permissive access roles. This study analyzes prominent security incidents alongside current research on cloud and AI threats to identify recurring weaknesses in configuration management, secret handling, and model integrity. The findings highlight how attackers exploit these gaps to steal data, engage in cryptojacking, and gain unauthorized access to AI models. To address these risks, the paper proposes a framework for fortifying AI infrastructure that emphasizes: (1) zero-trust identity and access management, (2) secure coding and model lifecycle practices, (3) automated configuration scanning, and (4) continuous policy enforcement. The results underscore that AI infrastructure should be treated as national critical infrastructure, warranting rigorous standards and proactive defense measures. Without systematic hardening, AI pipelines are high-value targets for cybercriminals and nation-state actors, posing a threat to public safety and national security.

Keywords: *AI Infrastructure, Cloud Security, Infrastructure as Code (IaC) Security, Code and Data Integrity, National Cybersecurity.*

1. INTRODUCTION

Artificial intelligence (AI) is increasingly integrated into national infrastructure, encompassing defense, healthcare, and energy systems. Governments and critical industries employ AI for tasks ranging from autonomous decision support to large-scale data analysis. To meet performance and scalability demands, many of these workloads are deployed on public cloud platforms such as Amazon Web Services (AWS) and Microsoft Azure (Malhotra & Massimi, 2024). Cloud platforms offer elastic computing power and specialized AI services that accelerate national projects. Still, they also introduce a shared responsibility model: while providers secure the underlying infrastructure, customers, including national agencies, must safeguard their applications, configurations, and data.

Despite provider guidance, insecure code and misconfigured cloud resources remain common in practice, often serving as the root cause of major breaches. Gartner has projected that through 2025, 99% of cloud security failures will result from customer misconfiguration (Malhotra & Massimi, 2024). Notable incidents illustrate the severity of this issue: Capital One's 2019 AWS firewall misconfiguration exposed personal information of more than 100 million individuals (Stella, 2019); attackers hijacked Tesla's cloud resources by exploiting an unsecured DevOps tool for cryptocurrency mining (Newman, 2018); and in 2023, an overly permissive Azure token exposed 38 terabytes of sensitive Microsoft AI data (Ben-Sasson & Greenberg, 2023). These examples demonstrate how a single exposed secret or misconfigured setting can compromise critical infrastructure.

The risks extend beyond organizational boundaries. In a national security context, adversaries exploiting vulnerabilities in AI systems used for energy grid management or military logistics could trigger large-scale disruption. Yet, no binding standards currently govern the security of AI pipelines. Many organizations continue to treat AI and machine learning (ML) projects as experimental initiatives rather than as essential infrastructure that requires rigorous controls (Arnold, 2025).

This research argues that fortifying AI infrastructure is not only a best practice but a national security imperative. To guide this analysis, the following research questions are posed:

- *RQ1*: **What are the common weaknesses in cloud-hosted AI infrastructure, spanning code, configuration, and model integrity, that adversaries exploit?**
- *RQ2*: **What national-level consequences could arise if these vulnerabilities are weaponized?**
- *RQ3*: **Which security frameworks and best practices are most effective for mitigating risks in major cloud platforms such as AWS and Azure?**

To address these questions, this study reviews the current landscape of cloud and AI security, analyzes representative breach case studies, and proposes a framework for enhancing the security of AI systems. The scope encompasses both technical controls (e.g., secure coding, configuration hardening, and model protection) and governance measures aligned with emerging national cybersecurity policies. By synthesizing academic research and industry reports, this work aims to inform policymakers and practitioners seeking to secure AI services that underpin critical national infrastructure.

2. LITERATURE REVIEW

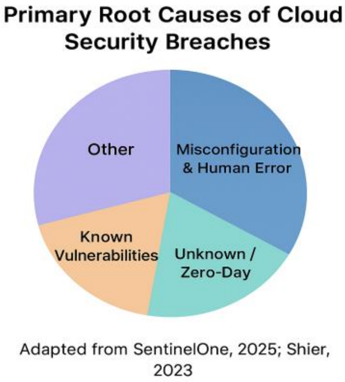
2.1 AI Infrastructure Lifecycle

Securing AI systems requires an understanding of the AI/ML lifecycle, which encompasses data ingestion, model training, deployment, and inference (Sham, 2024). Each stage introduces distinct risks. During data preparation and training, models may inherit errors or biases if datasets are corrupted or manipulated. In model development, data scientists and ML engineers construct pipelines using custom scripts, open-source libraries, and pre-trained models, often without formal secure coding practices. Trained models are then packaged, commonly in Docker containers, and deployed to cloud runtimes such as virtual machines, serverless functions, or Kubernetes clusters. The combination of traditional IT components with AI-specific elements expands the overall attack surface.

2.2 Weaknesses in cloud security

Cloud misconfiguration remains a leading cause of security incidents. Common issues include open storage buckets, excessive access privileges, and disabled logging (Solada, 2024). Surveys indicate that over 80% of organizations have experienced a cloud-related breach within 18 months, primarily due to misconfigured resources (Solada, 2024). Identity and access management (IAM) misconfigurations are especially critical, as overly broad permissions can escalate privileges or expose sensitive data (SentinelOne, 2025). Secret management is another persistent challenge: in 2023, GitHub detected 12.8 million exposed secrets, including API keys and TLS certificates, within public repositories (Toulas, 2024). Compromised credentials accounted for half of cyberattacks in the first half of 2023, surpassing vulnerability exploits (Shier, 2023).

Figure 1: Root causes of cloud security breaches. Misconfiguration and human error collectively account for nearly one-third of incidents, exceeding the exploitation of known vulnerabilities (adapted from industry reports). (SentinelOne, 2025; Shier, 2023)



Research is increasingly shifting toward vulnerabilities specific to AI/ML. Adversarial machine learning has demonstrated that models can be undermined through data poisoning or adversarial examples, degrading model integrity (NIST, 2023). However, these findings have yet to be fully adapted into cloud-native controls for AI deployments. Reviews highlight that few organizations systematically secure the entire ML pipeline, and existing cloud security benchmarks (e.g., CIS AWS and Azure) focus primarily on general misconfiguration rather than AI-specific risks such as dataset provenance or model tampering (Arnold, 2025).

2.3 Code and Configuration Integrity

Traditional software engineering employs well-established safeguards, such as static analysis, code signing, and supply chain vetting, but these are inconsistently applied in AI projects. Many ML workflows rely on scripts and Jupyter notebooks authored by data scientists without formal secure development training (OpenSSF, 2023). Scholars describe this as the "*MLOps security gap*," where rapid AI innovation outpaces governance and monitoring controls (Sham, 2024).

Infrastructure-as-Code (IaC) tools such as Terraform and CloudFormation increasingly provision AI workloads in the cloud. IaC scanning tools (e.g., Checkov, KICS, tfsec, and cfn-nag) enable early identification of insecure templates, such as public S3 buckets or permissive network ACLs (Wiz, 2024). Preventive measures (shift-left security testing) can be complemented by detective controls through configuration auditing and cloud security posture management (CSPM) systems, which continuously monitor deployed resources for drift or misconfiguration (Malhotra & Massimi, 2024). Yet, visibility remains limited: two-thirds of organizations report inadequate real-time insight into their cloud environments (Solada, 2024).

The literature indicates substantial difficulties in safeguarding AI infrastructure. Cloud deployments frequently suffer from misconfigurations, poor identity and secret management, and overlooked supply chain risks (SentinelOne, 2025; Toulas, 2024). Meanwhile, adversarial ML research emphasizes the need for data integrity and model robustness. However, few frameworks integrate these concerns into practical, cloud-native defenses. Building on this body of work, this study examines how such weaknesses manifest in real-world AI deployments on AWS and Azure and proposes a unified framework for mitigating these risks.

3. CASE STUDIES OF REAL-WORLD BREACHES

Analyzing security incidents provides practical insights into how theoretical vulnerabilities manifest in operational environments. This section examines three representative breaches: the 2019 Capital One data exposure on AWS, the 2018 Tesla Kubernetes cryptojacking incident, and the 2023 Microsoft AI model exposure on Azure. Together, these cases demonstrate how misconfigurations, weak secret management, and inadequate monitoring can compromise cloud-hosted AI systems.

3.1 Capital One AWS Breach (2019)

In July 2019, Capital One disclosed that a hacker had exfiltrated approximately 106 million customer records from AWS-hosted infrastructure (Capital One, 2019). The root cause was a misconfigured Web Application Firewall (WAF), which contained overly permissive rules. The attacker exploited this weakness using Server-Side Request Forgery (SSRF) to query the AWS metadata service, retrieving temporary credentials tied to an IAM role (Stella, 2019).

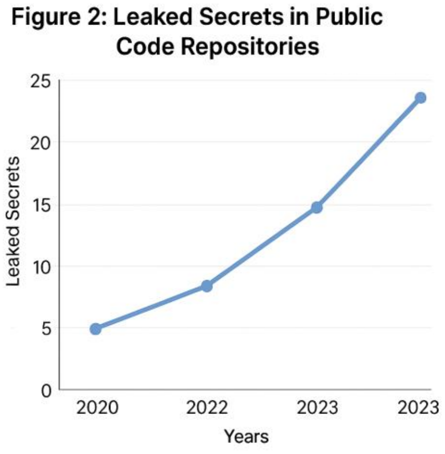
With these credentials, the attacker assumed excessive privileges that allowed the enumeration and download of hundreds of S3 buckets. The exfiltrated data included sensitive personal information, such as names, birth dates, Social Security numbers, and credit scores. The breach was discovered only after the attacker publicly disclosed the incident on social media. Subsequent investigations confirmed that a firewall misconfiguration combined with weak IAM scoping enabled large-scale data theft (Stella, 2019).

Lessons Learned:

- Enforce **least privilege IAM policies** to restrict role access to only essential data.
- Apply **strict SSRF mitigations**, such as AWS IMDSv2, to protect metadata services.
- Implement **real-time anomaly detection** for unusual S3 access and data exfiltration.
- Establish rigorous **change control and testing** for WAF rules and cloud configurations.

This case highlights how a single misconfiguration can escalate into a systemic compromise, particularly in environments that manage sensitive AI-driven workloads.

Figure 2: Millions of sensitive secrets (credentials, API keys, etc.) are accidentally leaked in public code repositories each year – a trend that has surged dramatically. This chart (GitGuardian data) shows the rising number of secrets detected on GitHub from 2020 to 2023, underscoring the prevalence of hardcoded credentials in code. (Toulas, 2024)



Millions of sensitive secrets (credentials, API keys, etc.) are accidentally leaked in public cole repositories each year – a trend that has surged dramatically. This chart (GitGuardan data) shows the rising number of secrets detected on GitHub from 2020 to 2023, underscoring the prevalence of hardcoded credentials in code. (Toulas, 2024)

3.2 Tesla Kubernetes Misconfiguration (2018)

In 2018, Tesla's AWS-based Kubernetes cluster was compromised due to an unsecured Kubernetes administration console (Newman, 2018). The web interface was left publicly accessible without authentication. Attackers leveraged this to access Kubernetes pods, where plaintext AWS credentials were stored in configuration files.

With these credentials, attackers pivoted into Tesla's broader cloud environment, installing cryptocurrency mining software on cloud servers. The malware consumed significant CPU resources while obfuscating network traffic through non-standard ports and protocols. Reports also suggested that compromised pods had access to telemetry data stored in S3 buckets, creating additional risk of internal data exposure.

Lessons Learned:

- Always enforce **authentication and role-based access control (RBAC)** for Kubernetes dashboards and management interfaces to ensure secure access.
- Avoid embedding cloud **credentials in containers or configmaps**; instead, use secure secrets management solutions.
- Monitor for **abnormal compute usage** (e.g., cryptojacking indicators) via cloud-native tools.
- Include orchestration tools (e.g., Kubernetes, monitoring dashboards) in **security baselines and audits**.

This incident highlights how misconfigurations in orchestration can expose AI infrastructure, as Kubernetes is widely used for deploying and scaling containerized models.

3.3 Microsoft AI Model Exposure (2023)

In 2023, researchers discovered that a Microsoft AI research team had inadvertently exposed 38 terabytes of internal data via an Azure Blob Storage Shared Access Signature (SAS) token (Ben-Sasson & Greenberg, 2023). Intended for distributing AI training datasets through a GitHub repository, the token was overly permissive, granting account-level access with full read, write, and delete permissions.

The exposed data included workstation backups containing internal Teams messages, source code, and secret keys. Critically, it also contained AI model checkpoints. Because the SAS token permitted write access, attackers could have modified or trojanized model files, potentially introducing backdoors into systems that downloaded them. Although Microsoft revoked the token and found no evidence of exploitation, the incident highlighted severe risks to model integrity and supply chain security (MSRC, 2023).

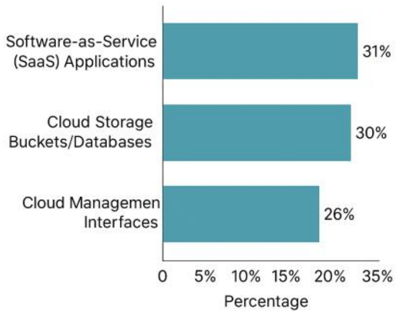
Lessons Learned:

- Restrict SAS tokens to a **minimal scope and short lifespans** to reduce the risk of misuse.
- Verify the integrity of distributed AI models using **checksums and digital signatures**.
- Employ **secret scanning and automated detection** to prevent token leakage in repositories.
- Treat **model files as sensitive assets**, requiring the same level of protection as source code.

This case highlights that model distribution channels represent a critical vulnerability in the security of AI infrastructure.

Figure 3: Key targets of cloud-focused cyber-attacks according to recent surveys. Software-as-a-Service (SaaS) applications (31%), cloud storage buckets/databases (30%), and cloud management interfaces (26%) are the most attacked assets in cloud environments (Spacelift, 2024; Newman, 2018). This frequently highlights that threat actors often seek target misconfigured storage or exposed management consoles as entry points. (Spacelift, 2024; Newman, 2018)

Figure 3: Key Targets of Cloud-Focused Cyber Attacks



Software-as-a-Service (SaaS) applications 31%), cloud storage buckets/databases (30%), and cloud management interfaces (26%) are the most commonly attacked assets in cloud environments (Spacelift, 2024; Newman, 2018). This emphasizes that threat actors often seek out misconfigured storage or exposed management consoles as entry points.

Across these incidents, a consistent set of failure patterns becomes clear. Each breach began with a seemingly minor misconfiguration, whether an overly permissive WAF rule, an exposed Kubernetes dashboard, or an improperly scoped SAS token, that provided an initial foothold for the attacker. These weaknesses were compounded by excessive privileges and inadequate secret management, which enabled access to expand far beyond what was initially intended. Furthermore, insufficient monitoring and delayed detection allowed adversaries to exploit these footholds on a large scale before response measures could take effect. Together, these cases demonstrate that cloud security lapses rarely stem from a single point of failure; instead, they result from compounding errors in configuration, identity and access management, and operational oversight. Addressing these challenges requires an integrated security approach that combines configuration hardening, least-privilege enforcement, and continuous monitoring practices tailored to the unique characteristics of AI environments.

4. ANATOMY OF AI INFRASTRUCTURE VULNERABILITIES

Building on the case studies, AI infrastructure vulnerabilities can be categorized into three main areas: **code-related flaws, cloud configuration weaknesses, and threats to model integrity**. While each class of vulnerability has unique causes and impacts, they often overlap in their effects. For example, insecure code artifacts may introduce a backdoor that is then amplified by a misconfiguration, granting excessive access. This section provides a detailed examination of each category.

4.1 Code Vulnerabilities in AI Pipelines

Hardcoded Credentials: A pervasive issue in AI and DevOps codebases is the embedding of secrets directly into notebooks, scripts, or configuration files. If repositories are exposed intentionally or accidentally, these secrets provide attackers with immediate access to sensitive resources. **GitGuardian** reported over 12 million leaked authentication secrets on GitHub in 2023, many of which remained valid for days (Toulas, 2024). In AI contexts, this could mean leaked dataset credentials, registry passwords, or even cloud API keys. The Capital One breach (2019) indirectly illustrated this risk, where AWS credentials were obtained via a misconfigured metadata service (Stella, 2019). Best practices recommend using centralized secret management systems (e.g., AWS Secrets Manager, Azure Key Vault) and avoiding the hardcoding of credentials in code or configuration files.

Lack of Code Signing and Verification: Unlike consumer software, AI models and pipeline scripts are rarely digitally signed. This creates opportunities for supply chain attacks; for instance, a compromised open-source ML library or tampered pre-trained model could silently propagate malicious behavior across dependent systems. The Microsoft SAS token exposure (2023) demonstrated how easily adversaries could have replaced legitimate model files with trojanized versions (Ben-Sasson & Greenberg, 2023). Mitigation includes artifact signing, checksum verification, and controlled registries (e.g., AWS CodeArtifact, Azure Artifacts).

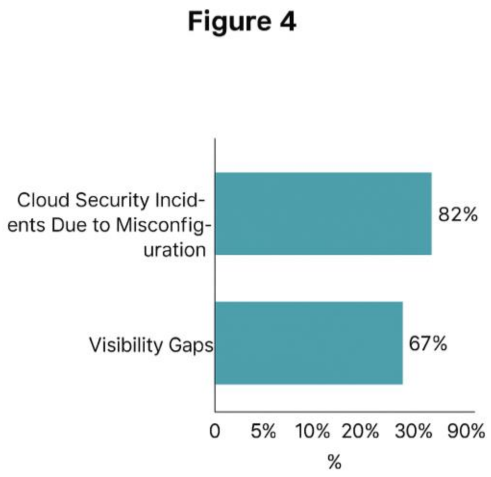
Risks from Open Source and Dependencies: Popular AI frameworks (TensorFlow, PyTorch) and community-driven code introduce supply chain risks, including typosquatting attacks and malicious package uploads. Python's ecosystem has repeatedly seen compromised packages that exfiltrate environment variables or secrets. Jupyter notebooks from untrusted sources pose similar risks. Organizations should integrate Software Composition Analysis (SCA), dependency scanning, and container security scanning (e.g., Trivy, JFrog Xray) into ML pipelines, ensuring workloads are isolated in hardened containers.

In short, code in AI pipelines must be secured as rigorously as production software. This involves integrating DevSecOps (Development Security Operations) practices, such as secret scanning, static analysis, artifact signing, and continuous dependency monitoring, into ML workflows (OpenSSF, 2023).

4.2 Configuration Vulnerabilities

A 2024 industry survey found 82% of enterprises experienced cloud security incidents due to misconfiguration and 67% struggled with visibility into their environments (Check Point, 2024).

Figure 4: 2024 survey showing 82% of enterprises had cloud security incidents due to misconfiguration, and 67% faced visibility gaps



IAM Misconfigurations: Identity and Access Management (IAM) is the backbone of cloud security. Overly broad IAM roles (e.g., AWS *AdministratorAccess*, Azure *Owner*) remain common, often due to convenience. The Capital One attacker exploited such a role to exfiltrate all their S3 buckets (Stella, 2019). Research by CloudKnox shows that **most identities use less than 5% of their assigned permissions**, highlighting widespread privilege bloat (SentinelOne, 2025). Mitigation strategies include least privilege by default, just-in-time privilege escalation (Azure AD Privileged Identity Management), and anomaly detection on unused or newly invoked permissions.

Cloud Storage Misconfigurations: Publicly exposed S3 buckets and Azure Blob containers remain a top attack vector (CSA, 2024). Risks include **data leakage, model tampering, or malicious data injection**. Cloud providers have improved defaults (S3 and Blob accounts are now private by default), but missteps still occur, particularly with legacy systems. Automated tools such as AWS Config and Azure Policy can continuously detect and remediate those insecure settings.

Default vs. Secure Defaults: Cloud tools often prioritize usability over security, which doesn't always work out. Tesla's 2018 breach highlighted how Kubernetes' unauthenticated dashboard was exploited for cryptojacking while legacy AWS and Azure services lacked default encryption or logging (Newman, 2018). To prevent these recurring patterns, organizations should enforce secure baselines via Infrastructure as Code (IaC) templates (Terraform, CloudFormation) with pre-approved guardrails (AWS Control Tower, Azure Blueprints).

The scale and complexity of cloud environments make human error inevitable. Mitigations require automation, policy-as-code, and secure baselines, not ad hoc manual reviews.

4.3 Integrity Threats to AI Models

Unlike traditional software, AI introduces new risks around model trustworthiness and resilience. Threats range from poisoning training pipelines to tampering with deployed models.

Model Poisoning Attacks: One primary concern is **model poisoning**, where attackers can inject malicious data into training pipelines, resulting in subtle yet dangerous misclassifications (e.g., mislabeling cancerous lesions as benign). In critical infrastructure, poisoned demand-forecasting models could destabilize power grids. Mitigation involves **tracking data provenance, detecting outliers, and maintaining audit trails** (CISA, 2023; NIST, 2023).

Adversarial Inputs (Evasion Attacks): At inference time, models face **adversarial input attacks**, where small, carefully designed perturbations cause misclassifications. Researchers have demonstrated that slightly altered stop signs can be misinterpreted as speed limit signs, potentially endangering autonomous vehicles. Defense strategies include **adversarial training, runtime anomaly detection, and red team testing** to stress-test models against adaptive threats (Arnold, 2025).

Model Theft and Exfiltration: Proprietary models represent high-value intellectual property that attackers can steal models directly from storage or reconstruct them through model extraction queries. The Microsoft SAS token exposure (2023) revealed how easily sensitive model files could be accessed at scale (Tramèr et al., 2016). Defenses include **role-based access control, model encryption, trusted execution environments (TEEs), and monitoring** for suspicious query patterns.

Model Tampering and Backdoors: Malicious actors may overwrite legitimate models with backdoored variants that behave erratically in response to specific trigger inputs. Organizations can guard against these through preventive controls, including **artifact signing, versioning, checksum validation, and staged deployment tests** (e.g., canary validation on known adversarial triggers).

In summary, AI models must be protected as critical assets, requiring both traditional security controls like IAM, encryption, monitoring, and AI-specific safeguards such as adversarial testing, model signing, and provenance tracking.

5. NATIONAL IMPLICATIONS OF AI VULNERABILITIES

The vulnerabilities discussed earlier are not only organizational challenges but also national security concerns. As AI systems become embedded in critical infrastructure, their insecurity translates into risks at a societal scale. Nation-state adversaries are increasingly targeting these systems, while the legal and policy frameworks that govern them remain underdeveloped. This section examines the implications of AI insecurity for critical infrastructure, the motivations and capabilities of threat actors, and the governance gaps that leave nations vulnerable to these threats.

5.1 AI in National Critical Infrastructure

AI technologies are becoming increasingly prevalent in the control and decision-support systems of critical infrastructure sectors.

Energy: AI models forecast electricity demand and optimize grid operations. Smart grids use AI to detect faults and automatically balance loads. A compromised model could misallocate resources or trigger cascading blackouts.

Healthcare: National health systems employ AI for diagnostics (e.g., radiology image analysis), hospital resource allocation, and outbreak prediction. Compromised systems could lead to misdiagnoses, treatment delays, or breakdowns in healthcare delivery.

Transportation: AI supports urban traffic optimization, autonomous transit, and air traffic management. Failures or tampering in these domains could cause accidents, reroute trains, or disrupt entire transit networks.

Defense: Militaries leverage AI for surveillance, logistics, cybersecurity, and strategic analysis. Manipulating or degrading these systems could erode battlefield awareness, disrupt supply chains, or enable adversarial deception. This weakens a country's strategic edge, leading to wrong tactical decisions.

Industrial and Manufacturing: AI manages predictive maintenance, robotic controls, and quality assurance in manufacturing, including defense and medical supply chains. A compromised system could cause downtime, sabotage production, or introduce flaws into critical components.

In each of these sectors, compromise has immediate and severe consequences. A poisoned model in an energy grid could mispredict loads and destabilize supply. An adversary tampering with AI-driven healthcare diagnostics could cause harm to patients. Real-world incidents already highlight these risks: ransomware campaigns have disrupted hospitals, and a water treatment facility's AI-enabled system was reportedly altered to change chemical dosing, with potentially fatal consequences if not detected (Dragomer, 2024).

The U.S. Cybersecurity and Infrastructure Security Agency (CISA) and other regulators emphasize that AI in critical infrastructure must be treated as a safety-critical system. Protecting model integrity, data provenance, and supply chain resilience is not merely a technical challenge but a matter of public safety and economic stability. This may require classifying specific AI applications as "critical systems," subject to the same oversight applied to the power grid or financial networks (CISA, 2023).

5.2 Threat Actor Motivations and Capabilities

Threats to AI systems originate from a range of actors, including cybercriminal groups and state-sponsored advanced persistent threats (APTs). Unlike opportunistic attacks, nation-state operations are driven by strategic objectives and backed by significant resources.

Espionage: State actors seek to exfiltrate AI models, training datasets, and semiconductor designs to accelerate domestic research and development (R&D) and gain economic or military advantages. In 2023, U.S. authorities indicted individuals tied to Chinese intelligence for targeting frontier AI research, confirming the scale of this effort (Arnold, 2025).

Sabotage and Disruption: Subtly degrading the accuracy or availability of AI systems can have significant and far-reaching effects; for example, disrupting military logistics, weakening AI-powered cybersecurity, or destabilizing smart grids. Microsoft reported in 2022 that APT groups were probing AI systems, likely testing them for offensive use (Infosecurity Europe, 2024).

Information Operations: AI-driven disinformation campaigns, including deepfakes and manipulated recommendation systems, allow adversaries to erode public trust and sow social division. For example, a country could utilize social media AI bots or recommendation algorithms to disseminate specific messages and cause trouble in another country discreetly. Compromising AI curation systems for news or social media could lead to systematic distortion of information flows (WEF, 2025).

Cybercrime: Financially motivated groups exploit AI weaknesses for profit, from targeting hospital AI systems with ransomware to using generative AI for phishing or fraud. One case involved criminals leveraging AI voice cloning to impersonate a CEO and authorize a $240,000 transfer (Skorup, 2025).

The capabilities of top-tier adversaries are formidable. Nation-states can mount **long-term supply chain attacks** by embedding backdoors into AI models or infrastructure during their development and deployment. Analysts warn of side-channel attacks to reconstruct model architectures and evidence of campaigns aimed at stealing proprietary model weights (Arnold, 2025). Such operations blur the line between traditional espionage and AI-specific exploitation.

In short, **AI vulnerabilities are national security vulnerabilities**. When adversaries with geopolitical motives target critical AI, the risks extend far beyond corporate IT losses, demanding a defensive posture built for **sophisticated, well-funded opponents**.

5.3 Legal and Policy Gaps

The rapid adoption of AI in critical sectors has outpaced the development of enforceable security regulations. While some industries, such as finance or energy, operate under established cybersecurity mandates, AI-specific security requirements remain largely absent.

The U.S. **NIST AI Risk Management Framework (AI RMF)**, released in 2023, offers voluntary guidance on resilience and trustworthiness (NIST, 2023). Similarly, sector regulators such as the FDA and FAA have begun to examine AI risks but lack comprehensive mandates. Current regulations often address only the effects of a breach (e.g., GDPR penalties for leaked data) rather than the root causes of insecure AI pipelines.

At the policy level, the White House's **AI Bill of Rights (2023)** and Executive Orders on AI security have raised awareness, and the DHS issued a framework for AI in critical infrastructure (DHS, 2024). Yet these remain advisory rather than binding. In contrast, the EU's forthcoming AI Act takes a more prescriptive approach, classifying "high-risk AI systems" and requiring conformity to security standards before deployment. If enacted, it could establish the first enforceable baseline for AI security in critical domains.

A persistent challenge is **accountability**. Unlike traditional IT systems, where liability frameworks are clearer, AI failures often raise questions about responsibility: is it the model developer, the deploying organization, or the regulator? In the absence of explicit liability for AI security lapses, organizations may prioritize meeting existing IT compliance requirements over investing in AI-specific protections.

Progress is emerging through industry-led efforts. The **Cloud Security Alliance (CSA)** has established AI security working groups, IEEE is drafting trustworthy AI standards, and the U.S. Department of Defense has integrated secure AI into its strategic plans (DoD, 2022). However, these efforts remain uneven and non-binding.

In practice, **AI security relies heavily on an organization's level of maturity and risk tolerance**. Large banks may apply stringent controls because of regulatory scrutiny, while smaller firms may delay investments until after an incident. Without consistent national or international standards, this uneven adoption creates systemic risk.

In summary, AI vulnerabilities have clear national implications. Critical infrastructure is increasingly dependent on AI, adversaries view it as a strategic target, and governance frameworks are lagging far behind. Until binding standards are developed, **nations remain exposed to cascading risks that combine technical fragility with geopolitical pressure**. Proactive measures, treating AI as safety-critical, embedding DevSecOps practices into national infrastructure, and establishing enforceable security baselines are essential to prevent reactive legislation following a major AI incident.

6. FRAMEWORK FOR SECURING AI INFRASTRUCTURE

Protecting AI infrastructure requires a **defense-in-depth approach** that addresses vulnerabilities across configurations, code, infrastructure, and governance. We propose a four-part framework: (1) secure-by-default cloud configurations, (2) secure code and model development practices, (3) Infrastructure-as-Code (IaC) scanning and automation, and (4) Policy-as-Code for continuous compliance. Together, these layers form a resilient security posture across the **platform, pipeline, and governance levels**.

6.1 Secure-by-Default Cloud Configurations

Cloud environments that host AI workloads must start from hardened defaults rather than reactive patching. This principle of **secure-by-default** reduces the attack surface before workloads are even deployed.

Identity and Access Management: Apply strict least-privilege principles. Role-based access control (RBAC) should be designed so that an AI training service can access only the data it requires, no more. Cloud-native tools such as AWS IAM Access Analyzer and Azure AD Privileged Identity Management help identify and remediate overbroad permissions. Multi-factor authentication (MFA) should be mandatory for all accounts with administrative or development access. Where possible, use ephemeral credentials (e.g., AWS IAM roles, Azure Managed Identities) instead of hardcoded keys.

Data Protection: Since AI pipelines consume large volumes of sensitive or proprietary data, storage security is paramount; therefore, encryption at rest and in transit should be enforced. Customer-managed keys should be utilized for regulated or sensitive datasets because they provide stronger control over rotation and revocation.

Network Security and Segmentation: Network design should prevent unnecessary exposure of AI services to the public internet. Placing workloads in private subnets with tightly controlled ingress/egress policies limits potential entry points for attackers. Production inference should be isolated from training environments and only connected through managed gateways or private links (e.g., AWS PrivateLink, Azure Private Endpoints). This segmentation reduces lateral movement in the event of a compromise.

Governance and Blueprints: Enforcing secure configurations at scale is challenging without automation. Services such as AWS Control Tower and Azure Blueprints enforce these guardrails, ensuring that new projects inherit the baseline protections (e.g., logging is enabled, public buckets are blocked, and disk encryption is mandated). This minimizes configuration drift across large, multi-account deployments.

Continuous Monitoring: Secure-by-default is not a static approach. Threat actors adapt, and environments change. Native tools such as AWS GuardDuty and Microsoft Defender for Cloud detect anomalous behaviors (e.g., unusual API calls, mass data exfiltration, cryptojacking). Periodic security audits against benchmarks, such as the CIS Cloud Security Controls, further reinforce compliance and help identify blind spots.

In short, a secure-by-default posture ensures that **misconfiguration vulnerabilities**, a leading cause of real-world AI infrastructure breaches, are minimized from the outset and continuously corrected over time.

6.2 Secure Code and Model Development Practices

Just as traditional applications must embed security into development, AI systems require secure practices across both **software code and machine learning artifacts**. Treating code and models as first-class assets ensures integrity and resilience against supply chain threats.

Code Signing and Integrity Checks: Ensuring that code, containers, and models are cryptographically signed during Continuous Integration/Continuous Deployment (CI/CD) prevents tampering. Hashes can be validated at deployment, ensuring that models used for inference are identical to those trained and tested.

Secrets Management: Many breaches stem from exposed credentials left in configuration files or code repositories. Centralized secret vaults, such as AWS Secrets Manager or Azure Key Vault, mitigate this by securely storing credentials, tokens, and keys and retrieving them at runtime, rather than embedding them in code or configuration files. Pre-commit hooks and CI scans can catch those hardcoded secrets before they leave a developer's workstation, preventing accidental leakage.

Supply Chain Security: AI development often relies on open-source libraries, container images, or pre-trained models. These dependencies are attractive targets for attackers, which is why they must be sourced from trusted registries, regularly updated, and scanned for vulnerabilities. Container security services from AWS, Azure, and other cloud providers can enforce vulnerability-free builds. Trojanized pre-trained models demonstrate the need to validate both the authenticity and behavior of downloaded artifacts.

Secure Model Development Lifecycle: Protecting training pipelines requires dataset provenance tracking, adversarial robustness testing, and controlled access to data. Practices such as adversarial training, watermarking, and restricted model permissions mitigate poisoning, theft, and misuse. If a production AI only needs to process the current input data and a specific model file, it shouldn't access the entire data lake or all previous training data unless required. This reduces the risk for attackers, even if they gain access to the model serving environment.

By integrating security directly into development, organizations can reduce their risk of supply chain attacks, malicious model injection, and data poisoning, threats that are particularly heightened in AI environments. Aligning these practices with modern DevSecOps norms while accounting for model-specific risks, reducing exposure to both traditional vulnerabilities and new AI-specific risks.

6.3 Infrastructure as Code (IaC) Scanning and Automation

IaC has become the foundation of modern cloud operations, enabling reproducible and automated infrastructure provisioning. For AI environments, this offers a powerful opportunity to enforce security controls as part of the build process. Below are the benefits of using IaC in AI infrastructure.

Automated Scanning: Tools such as Checkov, tfsec, and KICS detect insecure configurations (e.g., public S3 buckets, open security groups) before deployment. Integrating these checks into CI/CD pipelines shifts security left, ensuring that insecure configurations, like overly permissive network rules or unencrypted storage, are flagged before deployment. These issues are identified when they are most cost-effective to remediate, thereby reducing the number of risky changes that reach production.

Secure Deployment Templates: Pre-approved IaC modules act as secure building blocks for teams, ensuring consistency. For example, a hardened template module for an ML training cluster might enforce logging, encryption, and restricted access out of the box, preventing teams from deploying risky ad hoc configurations. This standardization accelerates development while maintaining a strong baseline.

Drift Detection: After deployment, infrastructure can drift from its intended state due to manual changes or updates. Detecting and remediating drift using tools like Terraform (terraform plan), AWS Config, or Azure Policy ensures that environments remain aligned with the intended security posture. Automated remediation, such as re-enabling encryption if it is disabled, reduces reliance on manual intervention.

In summary, by embedding IaC scanning and automation into engineering workflows, organizations transform infrastructure provisioning into a continuous security control, making security guardrails an integral part of development rather than an afterthought.

6.4 Policy-as-Code for Continuous Compliance

While IaC enforces build-time security, Policy-as-Code ensures that compliance is maintained dynamically across complex and rapidly changing AI environments. This approach elevates governance from periodic reviews to **real-time, programmatic enforcement**.

Declarative Policies: Tools like **Open Policy Agent (OPA)** and **HashiCorp Sentinel** enable organizations to encode compliance requirements in code, allowing for automated enforcement. Rules such as “all storage must be encrypted” or “no container may run as root” can be validated during builds or at runtime, blocking noncompliant deployments before they become risks.

Cloud-Native Enforcement: Major providers offer built-in guardrails, such as AWS Service Control Policies and Azure Policy, which can prevent dangerous actions at the cloud API level, including creating public buckets or disabling encryption. These controls complement IaC scanning by protecting against misconfigurations introduced outside the IaC workflow.

Auditing and Evidence: Policy engines produce detailed logs of every decision, creating an auditable trail for regulators and internal governance. Automated alerts and remediation workflows ensure that non-compliance is identified and addressed promptly, thereby reducing dwell time for high-risk changes.

Scalable Governance: Policy-as-Code empowers small security teams to govern large organizations by embedding compliance into development pipelines. Industry examples, such as Netflix’s “governance as code” model, demonstrate that this approach allows developers to innovate quickly while remaining within safe, automated guardrails.

In summary, Policy-as-Code ensures ongoing compliance in rapidly evolving AI environments, reducing dependence on manual oversight and meeting regulatory standards for auditable, enforceable controls. Policy-as-Code transforms compliance from a reactive, human-led process into a scalable, continuous control system, aligning AI infrastructure with both operational flexibility and regulatory demands.

Together, these four pillars — secure-by-default configurations, secure code and model development, IaC scanning, and Policy-as-Code form a layered defense strategy. Configurations provide a hardened foundation, development practices secure the AI pipeline, IaC enforces prevention and consistency at build time, and Policy-as-Code ensures continuous enforcement and compliance at runtime. This integration transforms security from a static checklist into a dynamic, adaptive system that scales with the complexity of AI ecosystems. For organizations treating AI as mission-critical infrastructure, such a holistic approach is essential to maintaining resilience against both current and emerging threats.

7. DISCUSSION

The proposed framework provides a structured approach to enhancing the security of cloud-hosted AI infrastructure; however, its adoption raises several practical challenges. A central tension lies in striking a balance between **security and flexibility**. AI innovation thrives on rapid experimentation, open-source adoption, and fast iteration, yet strict security controls can be perceived as obstacles. For instance, requiring security review for every new library or inserting scanning and signing steps into CI/CD pipelines may slow down deployments and frustrate researchers working in fast-paced environments.

Addressing this tension requires making security **as seamless and automated as possible**. Education is a critical enabler: when developers and data scientists understand why controls exist, they are more likely to design with security in mind rather than viewing it as a burden. Equally important is providing "paved road" solutions, approved templates, hardened base images, and secure infrastructure modules that make the secure path the easiest path. Embedding security engineers directly into AI teams (the **DevSecOps model**) can further ensure that controls are tailored to workflows, enabling security to support rather than stifle innovation.

Another challenge lies in the **limitations of current security tools**. Traditional cloud security posture management systems excel at catching misconfigurations but are not designed to detect AI-specific risks, such as adversarial examples, data poisoning, or model theft. While this framework recommends robustness testing and dataset validation, systematic methods for measuring and improving model resilience are still in the process of emerging. Interim approaches, such as sanity checks with traditional algorithms or monitoring for sudden shifts in model accuracy, can serve as stopgaps; however, scaling AI robustness into production remains an open area of research.

The evolving threat landscape further complicates defenses. As organizations adopt stronger baselines, such as code signing and IaC scanning, attackers may shift their focus toward targeting signing infrastructure, exploiting insider threats, or leveraging AI to create polymorphic malware and automate reconnaissance. This creates an "AI vs. AI" dynamic in cybersecurity (CISA, 2023). Defenders must be equally adaptive, employing AI-driven anomaly detection and threat intelligence to keep pace with adversaries.

Organizational alignment also plays a decisive role. Implementing the framework often requires new skill sets at the intersection of cloud engineering, data science, and security. Enterprises may need to establish dedicated **machine learning security teams** or appoint an AI security Lead to bridge silos. AI risk must also be incorporated into broader **enterprise risk management**, ensuring that executive leaders and risk officers understand AI threats alongside more traditional IT and operational risks.

Finally, **regulatory and compliance considerations** are becoming increasingly important. While current standards provide limited AI-specific guidance, this is expected to change in the future. Organizations that proactively adopt strong AI infrastructure security may not only be better positioned for future regulations but also gain a competitive advantage by demonstrating trustworthy AI practices to regulators, partners, and customers. Participating in industry initiatives, such as contributing to NIST's AI Risk Management Framework or CSA's AI security guidelines, can help shape practical standards while aligning with peers.

Even with preventive measures, residual risk remains. Organizations must prepare for the possibility of AI-related breaches by implementing robust incident response plans. This includes scenario-based exercises such as testing how operations would revert to manual control if an AI model in a critical system were compromised, or how investigators would trace decisions influenced by a manipulated model. Such exercises often reveal the need for additional monitoring of model performance, early-warning systems for accuracy degradation, and stronger collaboration across IT, data science, and operations teams.

In summary, securing AI infrastructure is not a one-time technical exercise but an ongoing process of balancing innovation with risk management. The framework provides a foundation, but its effectiveness ultimately depends on the organization's culture, its ability to continuously adapt to new threats, and its integration into the overall cybersecurity strategy.

8. CONCLUSION

AI systems are rapidly becoming integral to national infrastructure and critical services, yet their security has not kept pace with their importance. This paper examined how misconfigured cloud services, insecure development practices, and AI-specific vulnerabilities can escalate into incidents with national-level consequences. Case studies from Capital One, Tesla, and Microsoft illustrate how minor configuration errors, such as an exposed firewall port, an unsecured DevOps console, or an overprivileged access token, can trigger massive breaches. These incidents underscore that AI inherits both the risks of cloud computing and new attack vectors unique to machine learning pipelines.

A recurring theme is that many AI infrastructure failures stem from **fundamental cybersecurity hygiene issues, including** unmanaged identities, exposed secrets, inadequate monitoring, and misconfigured resources. These traditional lapses can have outsized consequences in AI environments, where attackers may not only exfiltrate data but also manipulate models, undermining decision-making in sensitive domains.

This research argued that AI infrastructure must be protected with the same rigor applied to other forms of critical infrastructure. We presented a framework centered on secure cloud configurations, secure code and model development practices, IaC scanning and automation, and Policy-as-Code for continuous compliance. Together, these measures can significantly reduce the attack surface. For example, automated IaC scanning and policy enforcement could have prevented misconfigurations exploited in the Capital One and Tesla breaches. At the same time, stronger code and model integrity controls could have mitigated risks similar to those experienced by Microsoft with its AI data exposure.

The key lessons are clear: cloud misconfigurations and unsafe AI code pose real and present dangers, but proactive measures can mitigate much of this risk. Organizations must adopt these practices early, rather than waiting for incidents to occur. Policymakers and industry bodies also have a role to play in treating AI infrastructure as a **national security priority**, establishing clearer standards, certifications, and audits akin to those governing other critical sectors.

Looking ahead, AI security remains a moving target. Future priorities should include the development of **national AI security standards**, the creation of an **AI incident database** to share lessons learned across sectors, and advances in **AI-specific defense techniques** such as automated anomaly detection of model outputs and formal verification methods for machine learning algorithms. As AI and cloud computing become increasingly intertwined, closer collaboration between AI researchers, engineers, and security professionals will be essential.

Ultimately, securing AI infrastructure is both a technical challenge and a strategic necessity. By safeguarding code, configuration, and model integrity today, we not only protect against present threats but also build the foundation of trust required for AI to reach its full potential in society.

9. DECLARATIONS

Availability of data and material

Not applicable

Competing interests

The author declares that there is no conflict of interest.

Funding

Not applicable

Author contribution

I, the author, contributed to this manuscript and approve this submission.

Acknowledgments

Not applicable

Authors' information

Author: Ifeoma Joy Eleweke

Current position: Software Engineer – DevOps Engineer

Author affiliations: Westcliff University

Corresponding author details: elewekeifeoma@gmail.com

REFERENCES

[1] Arnold, J. R. (2025). High Risk AI Models Need Military Grade Security. *War on the Rocks*, August 6, 2025.

[2] AWS. (2020). *AWS Control Tower – Best Practices for Multi-Account Security*. Amazon Web Services Whitepaper.

[3] Ben Sasson, H., & Greenberg, R. (2023). 38TB of data accidentally exposed by Microsoft AI researchers. *Wiz Blog*, September 18, 2023.

[4] Capital One. (2019). 2019 Capital One Cyber Incident – What Happened (Press Release), July 2019.

[5] CISA. (2023). *Securing AI/ML Systems: Guidance for Critical Infrastructure*. Cybersecurity & Infrastructure Security Agency, May 2023.

[6] DHS. (2024). *AI Safety and Security in Critical Infrastructure – Framework and Recommendations*. U.S. Department of Homeland Security, November 2024.

[7] HealthTech. (2025). The Biggest Healthcare Cybersecurity Threats in 2025. *HealthTech Magazine*.

[8] Malhotra, A., & Massimi, M. (2024). Cloud security evolution: Years of progress and challenges. *IBM Think Blog*, March 14, 2024.

[9] Microsoft Security Response Center (MSRC). (2023). Microsoft mitigated exposure of internal information in a storage account due to overly permissive SAS token. *MSRC Blog*, September 18, 2023.

[10] Newman, L. H. (2018). Hackers Hijacked Tesla's Cloud to Mine Cryptocurrency. *Wired*, Feb 20, 2018.

[11] NIST. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0).* National Institute of Standards and Technology, January 2023.

[12] OpenSSF. (2023). *OpenSSF AI/ML Security Working Group Whitepaper*. Open Source Security Foundation, 2023.

[13] SentinelOne. (2025). 50+ Cloud Security Statistics in 2025. *SentinelOne Cybersecurity Blog*, updated August 5, 2025.

[14] Sham, S. (2024). IaC Scanning: Concepts, Process, and Tools. *Wiz Academy Blog*, October 30, 2024.

[15] Shier, J. (2023). Time Keeps on Slippin': The 2023 Active Adversary Report for Tech Leaders. *Sophos News*, August 23, 2023.

[16] Solada, J. (2024). 2024 Cloud Security Report: Misconfigurations & Limited Visibility Plague Enterprises. *DuploCloud Blog*, July 17, 2024.

[17] Spacelift. (2024). 100+ Cloud Security Statistics for 2025. *Spacelift Blog*, 2024.

[18] Stella, J. (2019). A Technical Analysis of the Capital One Cloud Misconfiguration Breach. *Cloud Security Alliance Blog (Fugue)*, August 9, 2019.

[19] Toulas, B. (2024). Over 12 million auth secrets and keys leaked on GitHub in 2023. *BleepingComputer*, March 12, 2024.

[20] Wiz. (2024). IaC Scanning: Concepts, Process, and Tools (Academy Article). *Wiz*, 2024.