International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Segregation, segmentation and zero trust: Building secure dev and QA environments

Ranjan Kathuria *

Information Security, Rubrik. United States of America.

International Journal of Science and Research Archive, 2025, 16(01), 080-088

Publication history: Received on 25 May 2025; revised on 28 June 2025; accepted on 02 July 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.16.1.2009

Abstract

Modern software development increasingly depends on development (Dev) and quality assurance (QA) environments that closely replicate production systems to enable rapid, reliable testing and deployment. While these environments accelerate innovation and reduce time-to-market, their complexity and frequent changes can introduce significant security risks if not managed with equal rigor as production. Overlooking robust security practices in Dev and QA can expose organizations to data breaches, regulatory non-compliance, and operational disruptions, ultimately undermining brand trust and business continuity.

This research paper presents a resilience oriented approach for securing Dev and QA environments, emphasizing proactive risk management and architectural discipline. The proposed methodology advocates for strict isolation of Dev, QA, and production environments using dedicated Cloud accounts and VPCs with granular network controls. It further recommends enforcing Zero Trust governance through continuous authentication, just-in-time and least privilege access, and eliminating implicit trust within internal networks. The framework incorporates threat informed defense by applying MITRE ATT&CK tactics to harden CI/CD pipelines and ephemeral testing resources. Compliance driven controls are also integrated, aligning with the NIST Cybersecurity Framework to ensure synthetic test data supporting regulatory requirements such as GDPR and ISO 27001.

By adopting these principles, organizations can significantly reduce the attack surface of non-production environments while maintaining development agility. This work demonstrates that treating Dev and QA environments with the same resilience and security focus as production is essential for safeguarding the entire software development lifecycle.

Keywords: Resilience-Oriented; Zero Trust; MITRE ATT&CK; NIST CSF; DevSecOps

1 Introduction

In today's hyper competitive technology landscape, speed of development is a key differentiator for organizations striving to deliver innovative products and features to market. Rapid development cycles, agile methodologies, and continuous integration/delivery (CI/CD) pipelines have become the norm, enabling teams to iterate quickly and respond to evolving customer needs. To support this velocity, organizations rely on separate development, testing, and production environments. These isolated environments allow developers to experiment, build, and test new features independently, minimizing risk to production systems and improving overall productivity.

The benefits are clear: separate Dev/QA environments enable parallel development, thorough testing, and a stable production release process. Developers can collaborate efficiently, identify and resolve issues early, and maintain the reliability of the user experience. However, this acceleration comes with a significant challenge which is securing these Dev/QA environments. According to the Delphix 2024 State of Data Compliance and Security Report, 54% of large enterprises surveyed experienced a data breach or data theft involving sensitive data in non-production environments

* Corresponding author: Ranjan Kathuria

(such as development, testing, analytics) within the last two years [8]. As the volume and sensitivity of data in these environments increase, so does the risk of exposure, especially when security controls lag behind those enforced in production.

1.1 Problem Statement

Despite the operational and business advantages of maintaining separate environments, non-production systems are frequently overlooked in security strategies. Many organizations prioritize speed and functionality in Dev and QA, often at the expense of robust security controls. This results in misconfigured environments, weak access controls, unmasked sensitive data, and inconsistent compliance practices. The complexity of modern architectures, combined with rapid deployment cycles and the use of third-party integrations, expands the attack surface and makes comprehensive risk management increasingly difficult. As a result, non-production environments have become a favored target for attackers, leading to data breaches, regulatory violations, and operational disruptions that can undermine brand trust and business continuity.

Aim

The aim of this research paper is to establish clear security guardrails that should be followed when developing non-production or development environments. This study proposes a resilience-oriented design that incorporates the NIST Cybersecurity Framework (CSF), Zero Trust framework and MITRE ATT&CK for Dev/QA, among other industry standards [2, 3, 4]. The research will present a practical architectural model and actionable recommendations to help organizations secure Dev and QA environments, ensuring that innovation and speed do not come at the expense of data protection, compliance, or organizational resilience.

2 Literature Survey

Recent studies emphasize that Dev and QA environments, while critical for rapid software delivery, often lag behind production in security rigor, making them attractive targets for attackers. Common pitfalls include misconfigurations, excessive permissions, and the reuse of real production data, all of which can lead to unauthorized access and data breaches. Best practices highlighted in the literature call for strict environment isolation, robust access controls, and continuous integration of security throughout the development lifecycle. Frameworks such as DevSecOps and ISO 27002 stress the need for separation between development, testing, and production, as well as the enforcement of least privilege principles and secure coding standards.The consensus is clear: treating Dev and QA environments with the same discipline as production is essential to reduce risk, ensure compliance, and protect organizational reputation in an era of increasing cyber threats.

3 Problem Definition and Experimental Work

3.1 Frameworks for Securing Dev and QA Environments

To address the security challenges of non-production environments, this research integrates three industry leading frameworks: NIST Cybersecurity Framework (CSF), Zero Trust Architecture and MITRE ATT&CK for Cloud. Each framework provides complementary principles and actionable controls to harden Dev and QA environments against modern threats [9, 10, 11].

3.1.1 NIST Cybersecurity Framework (CSF)

The NIST CSF provides a structured approach to managing cybersecurity risks across five core functions:

- Identify: Inventory Dev/QA assets (code, data, secrets) and assess risks (e.g., exposed APIs, shared credentials).
- Protect: Implement synthetic test data generation, encryption (KMS etc), and RBAC to limit access to sensitive resources.
- Detect: Centralize logs (AWS CloudTrail, SIEM etc) to monitor for unauthorized activity in Dev/QA environments.
- Respond: Automate incident response (e.g., revoke access on anomalous behavior).
- Recover: Restore sanitized backups and conduct post mortems to improve resilience [9].

3.1.2 Zero Trust Architecture

Zero Trust operates on the principle of "never trust, always verify", eliminating implicit trust within internal networks. For Dev and QA environments, this means:

- Isolation: Segregate Dev, QA, and Prod into separate Cloud accounts and VPCs, enforcing strict network controls (security groups, NACLs) to prevent lateral movement. The Production VPCs should not be peered with Dev or QA VPCs.
- Continuous Authentication: Require multi-factor authentication (MFA) for all access to Dev/QA resources, including CI/CD pipelines and cloud consoles.
- Least-Privilege Access: Grant permissions only to specific roles (e.g., developers, QA engineers) for the minimum time required (just-in-time access).
- Device Trust: Ensure only managed, compliant devices can connect to Dev/QA environments [10].

3.1.3 MITRE ATT & CK for Cloud

MITRE ATT&CK maps adversarial tactics, techniques, and procedures (TTPs) specific to cloud environments. For Dev/QA:

- Credential Access (T1078): Harden secrets management (Secrets Manager Solution) to prevent credential theft from CI/CD pipelines.
- Lateral Movement (TA0008): Apply network micro-segmentation to isolate ephemeral testing resources (e.g., Kubernetes pods).
- Persistence (T1098): Audit IAM roles and service accounts in QA environments to detect over privileged identities.
- Defense Evasion (T1562): Monitor CI/CD workflows for malicious code injection via compromised dependencies [11].

3.2 Integration of Frameworks

Combining these frameworks creates a layered defense:

- Zero Trust ensures no implicit trust in Dev/QA networks.
- NIST CSF provides a compliance aligned structure for risk management.
- MITRE ATT&CK offers threat specific mitigations tailored to cloud-native environments.

For example, enforcing Zero Trust governance (MFA, least privilege) aligns with NIST CSF's "Protect" function, while MITRE ATT&CK informs threat detection rules for lateral movement in Dev environments.

3.3 Calculation Of Risk

As mentioned above in section 1.1, according to the Delphix 2024 State of Data Compliance and Security Report, 54% of large enterprises surveyed experienced a data breach or data theft involving sensitive data in non-production environments (such as development, testing, analytics) within the last two years [8].

Assume that there is a development environment where:-

- Development environment directly accessible via the internet with no network segmentation from production systems, enabling cross-environment communication.
- Database uses default credentials with no multi-factor authentication (MFA) enforcement.
- Zero encryption for data at rest or in transit.
- No asset inventory maintained.
- No formal Incident Response Plan for the development environment.

This configuration creates a high-risk attack surface where:

- Production systems are directly compromisable through dev environment breaches.
- Default credentials and missing MFA enable trivial unauthorized access.
- Sensitive data is fully exposed due to encryption gaps.
- Incident containment is impossible without defined response protocols.

This section quantifies the risk of a misconfigured Dev environment (publicly exposed API and database) using three frameworks: Zero Trust, NIST CSF, and MITRE ATT&CK [12, 13].

3.3.1 NIST Cyber Security Framework (CSF) Risk Calculation

Table 1 Risk Score Calculation based on NIST Framework

NIST CSF Function	Gap Identified	Risk Weight (1-10)
Identity	No Asset Inventory for Dev/QA	7
Protect	No Encryption for Data at Rest or Transit	9
Detect	No Logging / Monitoring	8
Respond	No Incident Response Plan	6
Recover	No Backups for Dev Data	5

Total Risk Score

$$((7 + 9 + 8 + 6 + 5) / 50) \times 100 = 70\% \text{ (High Risk)}$$

3.3.2 Zero Trust Risk Calculation

3.3.2.1 Formula

$$\text{Risk (Zero Trust)} = \text{Likelihood} \times \text{Impact}$$

3.3.2.2 Variables

3.3.2.2.1 Likelihood

- Dev Public API Exposure : 0.9 (80% chance of exploitation)
- No Authentication : 0.9 (90% chance of exploitation)
- Total Likelihood : 0.9 x 0.8 = 0.72

3.3.2.2.2 Impact

- Data Sensitivity : 8/10 (Since dev is exposed to the internet and there is no data segregation)
- Downtime cost: $500,000 (estimated recovery costs)
- Total Impact: 8 × 500,000 = $ 4,000,000

3.3.2.2.3 Risk Score

- Risk (Zero Trust) = 0.72 x 4,000,000 = $2, 880, 000

3.3.3 MITRE ATT&CK Risk Calculation

Table 2 Risk Score Calculation based on MITRE ATT&CK Framework

Tactic	Technique ID	Exploitability	Impact	Risk Score
Initial Access	Exposed API (T1190)	9	8	7.2
Exposed Credentials	Default Credentials (T1078)	10	9	9.0
Lateral Movement	Exploit Public Facing App (T1190)	8	7	5.6

3.3.3.1 Total Risk Score

$$((7.2 + 9.0 + 5.6) / 30) \times 100 = 73\% \text{ (High Risk)}$$

4 Proposed Design

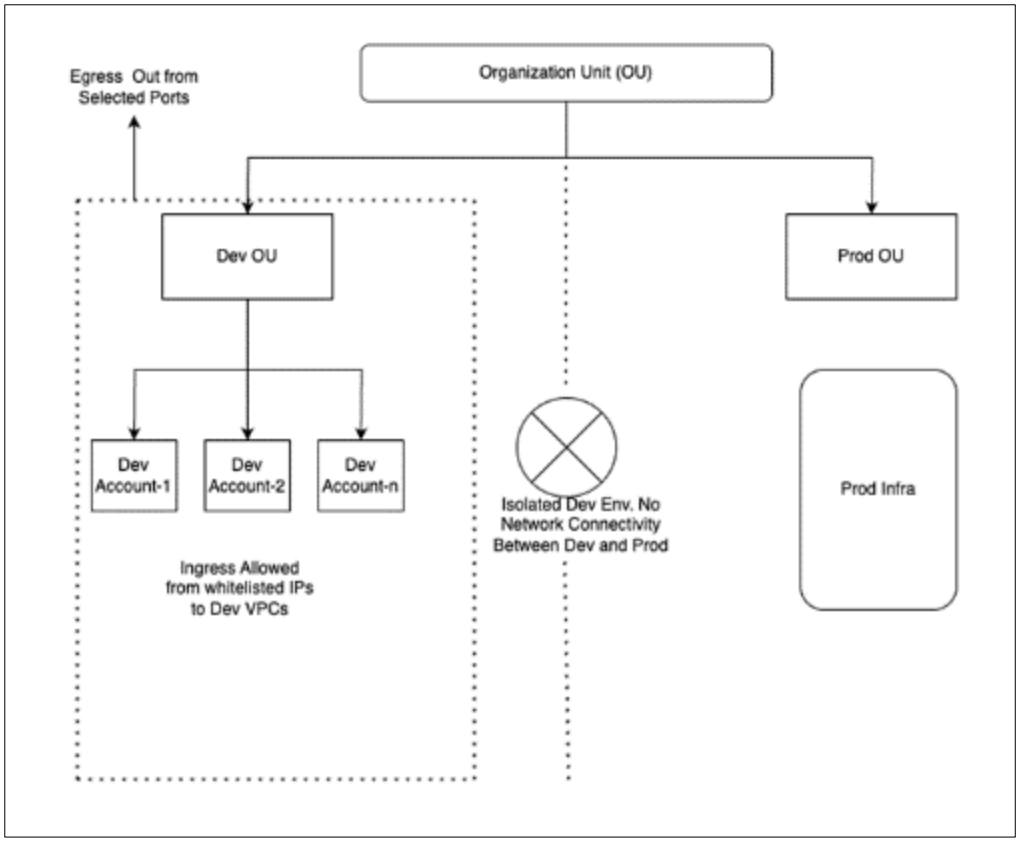

Figure 1 Secure Cloud Organization Unit Design

The design in "**Figure 1**" establishes rigorous segregation between development and production environments through three core mechanisms.

- Organizational Units (OUs) are strictly separated development and production cloud accounts reside in dedicated OUs with no overlap [14].
- A network boundary explicitly blocks all communication from development to production resources, preventing lateral movement.
- Development assets operate within an isolated Virtual Private Cloud (**VPC**) where ingress traffic is restricted to whitelisted IP addresses and egress is limited to specific authorized ports. This layered approach ensures complete isolation. Development environments cannot interact with production systems, while controlled network access minimizes external attack surfaces [15].

The architecture effectively contains risks within development zones and shields production from compromise vectors originating in less secure environments.

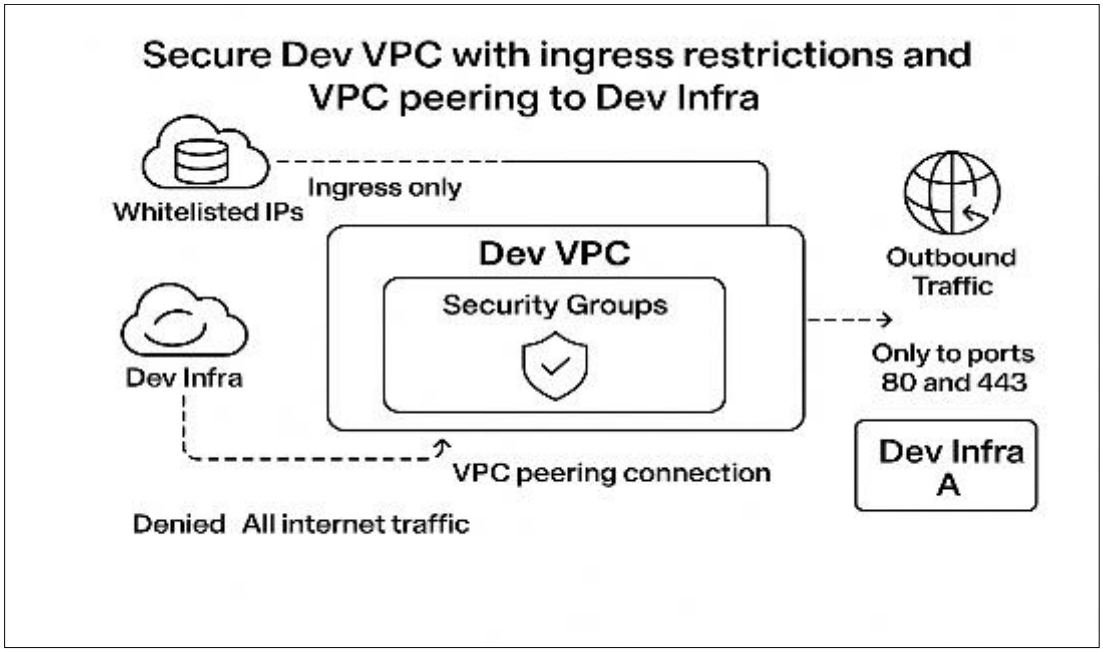

Figure 2 Segregation and Secure Design of Development VPCs

The design proposed in "**Figure 2**" enforces granular security within the Development Virtual Private Cloud (VPC) through three critical mechanisms.

- VPC peering is strictly limited to connections between Development Infrastructure and other Development VPCs, preventing unauthorized cross environment links.
- Ingress traffic is permitted exclusively from pre approved whitelisted IP addresses, eliminating exposure to unverified sources.
- Egress traffic is restricted to ports 80 (HTTP) and 443 (HTTPS), blocking all non web outbound communication by default or unless approved. This layered approach ensures development resources operate within a tightly controlled network perimeter shielding against external threats while containing internal traffic to essential web protocols.

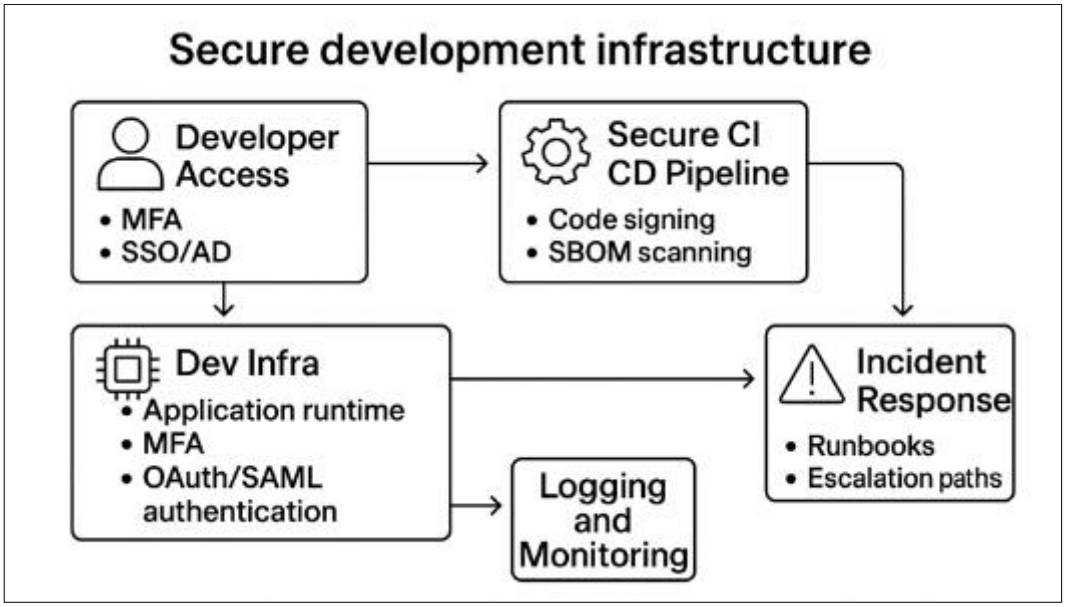

Figure 3 Design and Security Controls of Development Infrastructure

Building on the isolation provided by secure VPCs and Organizational Units in "Figure 1" and "Figure 2", the design presented in "**Figure 3**" outlines a comprehensive approach to developing, deploying, and accessing development infrastructure securely.

- Developers are required to authenticate using robust security measures such as Multi-Factor Authentication (MFA) and Single Sign-On (SSO), ensuring that only authorized personnel can access the development environment.
- The deployment process leverages a secure CI/CD pipeline, which incorporates code signing and Software Bill of Materials (SBOM) scanning to verify code integrity and detect vulnerabilities before deployment [16].

The environment is fortified with proactive security controls, including detailed security logging, continuous monitoring, and well-defined incident response playbooks.

5 Results and discussion

5.1 Risk Reduction

Building on the design proposed in Section 4, the solution features a secure development infrastructure that is fully isolated from the production environment and restricts access to only whitelisted IP addresses. Multi factor authentication is required for both authentication and application access, further strengthening security. The deployment process follows a secure model, and the environment is equipped with robust logging, monitoring, and an incident response playbook to ensure rapid detection and remediation of potential threats. Given the implementation of these controls, a re-evaluation of risk exposure is warranted based on the enhanced design specifications

5.1.1 Risk Reduction Based on NIST Cyber Security Framework (CSF)

Initial Risk: 70% (High)

- Encryption reduced the "Protect" risk weight from 9 to 4 (55.5% reduction).
- Logging/Monitoring reduced the "Detect" risk weight from 8 to 3 (62.5% reduction).
- Incident Response Playbooks reduced the "Respond" risk weight from 6 to 4 (33.3% reduction).
- Isolation reduced the "Recover" risk weight from 5 to 4 (20% reduction).
    - Residual Risk = ((7 + 4 + 3 + 4 + 4) / 50) x 100 = **44%**

5.1.2 Risk Reduction Based on Zero Trust Framework

Initial Risk: **$2, 880, 000**

- MFA reduced attack likelihood from 0.90 to 0.25 (72.2% reduction).
- Isolation reduced breach impact from 1M to 400K (60% reduction).
    - Residual Risk = Likelihood × Impact = 0.25 × 400,000 = $100,000

5.1.3 Risk Reduction Based on MITRE ATT&CK Framework

Initial Risk: 73% (Critical)

- MFA and Isolation mitigated:
    - Valid Accounts (T1078): Risk 10 to 3 (70% reduction).
    - Exploit Public-Facing App (T1190): Risk 8 to 1 (87.5% reduction).
- Logging Mitigated
    - Defense Evasion (T1562): Risk 7 to 2 (71.4% reduction).
    - Residual Risk = ((3 + 1 + 2) / 30) x 100 = **20%**

5.2 Risk Comparison & Results

Table 3 Risk Score Comparison based on Different Frameworks

Framework	Initial Risk	Residual Risk	Risk Reduction	Security Config Changed
NIST CSF	70% (High)	44% (Medium Risk)	37%	Encryption, Logging/Monitoring, Incident Response, Isolation
Zero Trust	$2, 880, 000	$100,000	$2,780,000	MFA, Least Privilege, Network Segmentation/Isolation
MITRE Attack	73% (Critical)	20% (Low)	53%	MFA, Isolation, Logging, Credential/Access Hardening

6 Conclusion

This research paper demonstrates, through practical analysis and quantitative risk modeling, that applying comprehensive security controls such as MFA, encryption, logging, incident response, and network isolation can substantially reduce risk in Dev and QA environments. By integrating industry frameworks like NIST CSF, Zero Trust, and MITRE ATT&CK, the study shows that organizations can lower their exposure to security threats by more than two thirds. The findings highlight that treating non-production environments with the same rigor as production is essential for protecting sensitive data, ensuring compliance, and maintaining organizational resilience in today's fast-paced development landscape

Compliance with ethical standards

Disclosure of conflict of interest

There are no conflicts of interest to declare.

References

[1] Grig Duta. Best Practices for Dev, QA, and Production Environments [Internet]. Bunnyshell. Available from https://www.bunnyshell.com/blog/best-practices-for-dev-qa-and-production-environments/.

[2] Scott Rose, Oliver Borchert, Stu Mitchell, Sean Connelly. Zero Trust Architecture [Internet]. NIST SP 800-207. Available from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf.

[3] MITRE ATT&CK for Cloud [Internet]. MITRE. Available from https://attack.mitre.org/techniques/enterprise/cloud/.

[4] Mitch Ashley. What is the NIST Cybersecurity Framework? [Internet]. DevOps.com. Available from https://devops.com/what-is-the-nist-cybersecurity-framework/.

[5] Synthetic Data Generation Best Practices [Internet]. Delphix. Available from https://www.delphix.com/glossary/synthetic-data-generation & https://www.perforce.com/blog.

[6] AWS Well-Architected Framework [Internet]. Security Pillar. Available from https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html.

[7] Ranjan Kathuria. Beyond Production: Why Securing Dev and QA Environments Matters [Internet]. Cyber Defense e-Magazines June 2025 Edition. Available from https://cyberdefensemagazine.tradepub.com/free/w_cyba177/.

[8] Delphix's State of Data Compliance and Security Report Reveals 54% of Organizations Experienced Data Breach in Non-Production Environments [Internet]. Available from https://www.perforce.com/press-releases/delphix-state-data-compliance-and-security-report-reveals.

[9] NIST Cybersecurity Framework [Internet]. Novatech. Available from https://novatech.net/blog/understanding-the-nist-cybersecurity-framework.

[10] Zero Trust Architecture [Internet]. Okta. Available from https://www.okta.com/identity-101/zero-trust-framework-a-comprehensive-modern-security-model/.

[11] Stefano Chierici. MITRE ATT&CK for Cloud [Internet]. Sysdig. Available from https://sysdig.com/blog/what-is-mitre-attck-for-cloud-iaas/.

[12] Rebecca Kappel. Risk Calculation [Internet]. Centraleyes. Available from https://www.centraleyes.com/7-methods-for-calculating-cybersecurity-risk-scores/.

[13] MITRE ATT&CK [Internet]. MITRE. Available from https://attack.mitre.org/.

[14] Managing Organization Unit [Internet]. Amazon Web Services. Available from https://docs.aws.amazon.com/organizations/latest/userguide/orgs_manage_ous.html

[15] What is Amazon VPC [Internet]. Amazon Web Services. Available from https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html.

[16] Gui Alvarenga. What is SBOM [Internet]. Crowdstrike. Available from https://www.crowdstrike.com/en-us/cybersecurity-101/exposure-management/software-bill-of-materials-sbom/.