ManageEngine
ADAudit Plus

7 Azure & AWS misconfigurations

to fix in your network

www.adauditplus.com

7 Azure & 7 AWS misconfigurations to fix in your network

Azure and AWS are critical for managing identity and access in the cloud. Understanding the security implications cloud platforms may pose, the platforms contain robust tools for securing your cloud infrastructure. However, misconfigurations within these tools can introduce significant risks. These vulnerabilities can expose sensitive resources, data, and applications to unauthorized access, leading to potential security breaches, data loss, or costly downtime.

Azure misconfigurations are often the result of oversights, misunderstandings of best practices, or simply the complexity of managing cloud environments at scale. Misconfigurations can occur in various areas, including network security, access controls, and service configurations, making it imperative for organizations to continually assess and address potential risks.

Misconfigurations in AWS environments, whether it's leaving sensitive services publicly accessible or failing to implement recommended security practices, can expose critical systems to unauthorized access, data breaches, and costly downtime. By addressing and fixing these misconfigurations, organizations can bolster their security posture, ensure compliance with best practices, and mitigate potential threats. Regularly reviewing and remediating these issues is essential for maintaining a secure AWS environment and safeguarding sensitive data.

In this ebook, we will explore common Azure and AWS misconfigurations and provide actionable steps to remedy these issues. By implementing these best practices, you can significantly reduce your security posture risks and safeguard your Azure and AWS environments.

7 Azure misconfigurations to fix in your network

S.No	Policy	Problem	Remedial Action
1	**Azure Front Door Has No Associated WAF Policy**	Without a Web Application Firewall (WAF), Azure Front Door is exposed to attacks such as SQL injection, XSS, and CSRF, potentially leading to data breaches, service disruptions, and unauthorized access to backend services.	1. Go to Azure Portal → Front Doors under content delivery. 2. Select your Front Door instance. 3. Attach a WAF policy from the 'Web application firewall' settings. 4. Apply and save changes.

2	**Azure SQL Server Firewall Rule Allows Access to All Public IPv4 Addresses**	Allowing SQL Server access from any IP address (0.0.0.0/0) is a security risk, as attackers can exploit vulnerabilities to gain unauthorized access to databases and cause data breaches.	1. Go to Azure Portal → SQL Servers. 2. Identify the firewall rule with start IP 0.0.0.0 and end IP 255.255.255.255. 3. Delete the rule or replace it with specific trusted IP ranges. 4. Save changes.
3	**LDAP, RDP, SNMP, and all other protocols allowing external traffic**	Exposing LDAP (port 389), RDP (port 3389), SNMP (port 161), and other protocols to external traffic can lead to attacks, including brute-force login attempts, unauthorized access, and sensitive data exposure.	1. Navigate to Azure portal → All Services → Network security groups. 2. Review inbound rules for ports 3389, 389, and 161. 3. Limit access to trusted IPs or block all public access. 4. To restrict inbound rules for any protocol, review rules with 'Any' protocol and 'Any' source. 5. Restrict the source to specific IPs or delete the rule. 6. Save changes.
4	**ElasticSearch Port is Publicly Accessible**	Exposing Elasticsearch ports (9200 or 9300) to the internet can allow attackers to steal, manipulate, or corrupt sensitive data, leading to data breaches and application attacks.	1. Review your NSG for rules allowing traffic to ports 9200 and 9300. 2. Navigate to Azure portal → All Services → Network security groups. 3. Identify rules for ports 9200 or 9300. 4. Restrict access to trusted IPs or deny all public access. 5. Save changes.
5	**Web App Uses Minimum TLS Version**	Using outdated TLS versions (e.g., TLS 1.0 or 1.1) makes web applications vulnerable to MITM, eavesdropping, and data tampering attacks, risking sensitive data such as user credentials and financial transactions.	1. Navigate to Azure Portal → Web Apps. 2. Under Configuration → General settings, set Minimum Inbound TLS Version to 1.2. 3. Save changes.

6	**Azure Key Vault Allows Traffic from All Networks**	Allowing public access to Azure Key Vault increases the risk of unauthorized access to sensitive information (API keys, passwords), leading to data breaches and service compromises.	1. 1. Go to Azure Portal → All Services → Key Vault. 2. 2. Under Networking, select Firewalls and virtual networks, and disable public access. 3. 3. Restrict access to trusted networks. 4. 4. Save changes.
7	**Azure VM Snapshot is Publicly Accessible**	Publicly accessible VM snapshots can contain sensitive data like encryption keys or personal information. If accessed by an attacker, these snapshots could lead to data theft or VM compromise.	1. 1. Navigate to Azure Portal → Snapshots. 2. 2. Check the network access settings for your snapshots. 3. 3. Disable public access or limit to trusted networks. 4. 4. Save changes.

7 AWS misconfigurations to fix in your network

S.No	Policy	Problem description	Remedial Action
1	**AWS EC2 instance not configured with IMDSv2**	EC2 instances without IMDSv2 are vulnerable to metadata service attacks, which can lead to unauthorized access to sensitive instance data like IAM credentials. IMDSv2 mitigates these risks by requiring session-based authentication.	1. Open the service console AWS EC2 Console 2. Select your region and instance ID. 3. Click "Actions" and select "Modify instance metadata options." 4. Enable "Instance metadata service" and check the "IMDSv2 Required" box. 5. Click "Save."

2	**Root account not configured with MFA**	Without MFA on the root account, a compromised root password grants full access to all resources, making the account a prime target for attackers. MFA provides an additional layer of security.	1. Sign in to AWS IAM Console as the root user. 2. Select your account on the right and click "Security credentials." 3. Expand the "Multi-factor authentication (MFA)" section and choose "Assign MFA device." 4. Follow the wizard to add a hardware TOTP token and enter the MFA device's serial number and codes. 5. Click "Add MFA" to complete the setup.
3	**RDS publicly accessible**	Publicly accessible RDS instances are exposed to the internet, increasing the risk of unauthorized access, brute-force attacks, and SQL injection. Best practices recommend private access unless there's a valid need for public access.	1. Go to AWS RDS Console. 2. Select the region and DB instance under "Resources." 3. Click on the DB Identifier and then "Modify." 4. Under "Connectivity," select "Not Publicly accessible." 5. Apply changes immediately by selecting "Apply immediately" and then "Modify DB Instance."
4	**Lambda function with admin privilege access**	Granting Lambda functions admin privileges increases the risk of inadvertent or malicious actions that could compromise your AWS environment. Adhering to least privilege is essential.	1. Log in to AWS Lambda Console. 2. Select the function and go to the "Configuration" tab. 3. Under "Permissions," click "Edit" to change the execution role. 4. Either associate an existing compliant role or create a new execution role with only necessary permissions. 5. Click "Save" to apply the changes.
5	**AWS EBS snapshots are accessible to public**	Public EBS snapshots can expose sensitive data to unauthorized users, potentially leading to data leaks or theft. Ensuring snapshots are private is critical for data security.	1. Go to AWS EC2 Console. 2. Select your region and click "Snapshots" under the Elastic Block Store section. 3. Choose "Actions" and then "Modify permissions." 4. Set "Snapshot availability" to "Private." 5. Click "Save changes."

6	**AWS S3 Buckets Do Not Have Server-Side Encryption**	Without server-side encryption, sensitive data in S3 buckets is stored in plaintext and vulnerable to unauthorized access, risking data leaks and compliance violations.	1. Go to AWS S3 Console. 2. Select the misconfigured bucket. 3. Go to the "Properties" tab and select the "Default encryption" section. 4. Choose either AES-256 or AWS-KMS for encryption. 5. For more info, refer to Default encryption and Policy based encryption.
7	**S3 Bucket Does Not Block Public Access**	When public access is not blocked, S3 buckets may unintentionally expose data to the public, risking unauthorized access and data breaches. Enabling "Block Public Access" is essential for protecting sensitive information.	1. Go to AWS S3 Console. 2. Select the misconfigured bucket. 3. Under "Permissions," find "Block public access (bucket settings)." 4. Click "Edit" and select the "Block all public access" checkbox. 5. Click "Save changes" and then confirm.

ManageEngine
ADAudit Plus

ADAudit Plus is a unified auditing solution that provides full visibility into activities across Active Directory (AD), Entra ID, file servers (Windows, NetApp, EMC and more), Windows servers and workstations—all in just a few clicks. ADAudit Plus helps organizations streamline auditing, demonstrate compliance and enhance their identity threat detection and response with capabilities like real-time change auditing, user logon tracking, account lockout analysis, privileged user monitoring, file auditing, compliance reporting, attack surface analysis (for AD, Azure, AWS, and GCP), UBA, response automation and AD backup and recovery. For more information about ADAudit Plus, visit www.manageengine.com/products/active-directory-audit/.

$ Get Quote

↓ Download