Pipeline Security Report

Analysis document

1. Introduction

OpenRemote is a provider of an open-source IoT platform, available both on GitHub and the AWS Marketplace. The company offers its clients the ability to deploy and manage the IoT platform through an AWS-based pipeline, which handles both the infrastructure setup and the platform's software deployment. Depending on the client's needs, OpenRemote either manages the entire infrastructure or provides the tools for clients to handle it themselves.

In response to growing demand for Azure-based solutions, OpenRemote is developing a similar deployment pipeline for Microsoft Azure. This project is a direct replication of their AWS pipeline, adapted to the Azure environment. The goal is to enable OpenRemote to extend its offerings to clients who prefer Azure over AWS for their infrastructure needs.

The first version of the Azure pipeline has been developed, leveraging Terraform to provision and manage the required resources. This security report focuses on assessing the security of this pipeline and its underlying Azure infrastructure. The primary goal is to identify vulnerabilities and propose enterprise-level improvements to ensure the pipeline meets the highest security standards.

2. Current Architecture

The Azure deployment pipeline for OpenRemote is built using Terraform to provision and manage infrastructure, with GitHub Actions orchestrating the deployment process. The current architecture includes:

- **Resource Group:** Provides centralised management, grouping related resources such as Virtual Machines, Storage Accounts, and networking for easier monitoring, billing, and lifecycle management.
- **Networking**: The networking layer is built using a Virtual Network with subnets for resource segmentation. Network Security Groups enforce strict access controls, regulating both inbound and outbound traffic. SSH access to Virtual Machines is restricted to an IP whitelist, as specified in the pipeline's configuration.
- **Virtual Machine:** Hosts the OpenRemote IoT platform, installed and configured using custom data. It is deployed within the Virtual Network to ensure secure communication with other resources.
- **Monitoring:** Optionally enabled to track metrics, such as CPU usage, and send alerts to a specified email address in case of high resource consumption.
- **Backup system**: An optional feature of the pipeline, providing automated point-in-time recovery for Virtual Machine disks and critical data to ensure business continuity in the event of accidental data loss or disasters.
- **Storage Account:** Used to securely store Terraform state files in a remote location.

The deployment process is initiated through a GitHub Actions workflow. The pipeline includes steps for getting user inputs, Terraform initialisation, validation, and application of configurations. Authentication with Azure is handled via OpenID Connect (OIDC), removing the need for static credentials and enhancing security. Static code analysis using Checkov ensures compliance with Terraform security best practices before deployment.

This architecture represents the initial implementation of the pipeline and provides a foundation for further analysis and improvement. The following sections will explore potential vulnerabilities and propose security enhancements to align the pipeline with industry standards.

3. Security Assessment

3.1. Scope

This security assessment evaluates the key components of OpenRemote's Azure deployment pipeline, focusing on areas critical to protecting the infrastructure and maintaining operational integrity. The following areas have been identified as essential to assess and improve the overall security posture of the pipeline:

- **Identity and Access Management:** Permissions and roles assigned to users, groups, and service principals are analysed to ensure they follow the principle of least privilege. Particular attention is given to identifying over-permissioned roles and ensuring access is appropriately scoped.
- **Network Security:** Virtual Network and subnet configurations are reviewed to ensure resource isolation. The effectiveness of Network Security Group rules is assessed to verify they restrict traffic appropriately, including SSH access through IP whitelisting.
- **Data Protection**: The storage of Terraform state files in Azure Storage Accounts is evaluated for secure access controls and encryption. Application data handling is also reviewed to ensure protection against unauthorised access and accidental exposure.
- **Monitoring and Backups:** Monitoring and alerting configurations are reviewed to ensure timely detection of unusual activity or failures. Backup strategies for critical resources, such as Virtual Machines and storage, are assessed to verify their reliability and adequacy.

3.2. Potential Risks and Recommendations

The assessment of risks aligns with the **NIST Cybersecurity Framework (CSF)**, referencing its core functions - **Identify, Protect, Detect, Respond, and Recover**, to ensure risks are identified and addressed using established best practices.

1. Identity and Access Management

The pipeline configuration assigns broad Contributor permissions at the Resource Group level, violating the principle of least privilege. This over-permissioning allows full control over resources, increasing the risk of exploitation if credentials are compromised. Additionally, the absence of a dedicated administrator account reduces accountability for administrative actions. Without enforced Multi-Factor Authentication, privileged accounts remain vulnerable to phishing and brute-force attacks. This issue aligns with the following NIST Cybersecurity Framework functions:

- **Identify (ID): ID.AM-05** highlights the importance of defining access permissions and managing them based on the principle of least privilege.
- **Protect (PR): PR.AA-05** stresses that permissions and authorisations must enforce the least privilege and separation of duties.

My recommendation is to replace Contributor roles with scoped permissions using Azure Managed Identities, establish a dedicated administrator account and enforce MFA for all privileged users to strengthen account security.

2. Network Security

The Virtual Machine hosting the IoT platform is configured with a public IP, unnecessarily exposing it to external threats. While Network Security Groups restrict inbound traffic, overly broad rules, such as 0.0.0.0/32, increase the risk of unauthorised access. HTTP/HTTPS traffic is unrestricted, leaving the platform open to web-based attacks. MQTT and SMTP traffic are also exposed, further increasing vulnerability of the virtual machine. SSH access is always on, relying on IP whitelisting, which is exploitable if misconfigured. Lastly, the absence of Azure DDoS Protection leaves the environment vulnerable to high-volume traffic disruptions. This issue aligns with the following NIST Cybersecurity Framework functions:

- **Identify (ID): ID.RA-01** advises identifying vulnerabilities in assets, such as exposed public IPs or open ports.
- **Protect (PR): PR.IR-01** recommends protecting networks and environments from unauthorised logical access, while PR.PS-05 highlights the need to prevent unauthorised protocols or services.

My recommendation is to place the Virtual Machine in a private subnet and remove its public IP. Route HTTP, HTTPS, MQTT and SMTP traffic through an Azure Load Balancer or Application Gateway. Enforce NSGs to restrict access to trusted IP ranges. For MQTT communication, ensure TLS encryption on port 8883 and mandate certificate-based authentication for IoT devices. Replace always-on SSH access with Just-in-Time (JIT) access or Azure Bastion, and enable Azure DDoS Protection for enhanced resilience.

3. Data Protection

Terraform state files are stored in Azure Storage Accounts without confirmed encryption or private endpoints. These files often contain sensitive metadata, such as resource IDs and credentials, making them a critical target for attackers. Additionally, static access keys for the Storage Account grant full privileges, increasing the risk of unauthorised access or data tampering if the keys are compromised. This issue aligns with the following NIST Cybersecurity Framework functions:

- **Identify (ID): ID.AM-01** underlines the importance of identifying and maintaining inventories of critical data and storage assets.
- **Protect (PR): PR.DS-01** focuses on protecting data confidentiality and integrity at rest through encryption and secure access controls.

My recommendation is to enable encryption at rest for Terraform state files and configure private endpoints to restrict access to internal Virtual Networks. Replace static access keys with Azure Managed Identities for secure and granular authentication.

4. Monitoring and Backups

Monitoring is currently limited to CPU usage, which is insufficient to detect critical events such as failed deployments, unauthorised access attempts, or resource modifications. Without comprehensive monitoring, incidents may go undetected, increasing downtime and recovery complexity. Backup configurations are optional, relying on user activation, which raises the risk of critical data loss if backups are not consistently enabled for resources like Virtual Machines. This issue aligns with the following NIST Cybersecurity Framework functions:

- **Detect (DE): DE.CM-01** highlights the need for monitoring networks and systems to detect adverse events.
- **Recover (RC): RC.RP-03** stresses the importance of validating the integrity of backups to ensure effective recovery after incidents.

My recommendation is to expand monitoring to include resource changes, unauthorised access attempts, and traffic anomalies using Azure Monitor and Log Analytics. Configure mandatory backups for all critical resources, using periodic snapshots stored in geo-redundant storage for reliability.

3.3. High-level Implementation Plan

The recommendations provided in this report form the foundation of the security enhancements for OpenRemote's Azure deployment pipeline. Implementation will prioritise critical risks, such as over-permissioning, public IP exposure, and secrets management, with actions focusing on the following key areas:

- **IAM and Access Control:** Enforcing least privilege, implementing MFA, and replacing broad roles with scoped permissions.
- **Network:** Putting Virtual Machine in a private subnet and route the traffic through load balancer.
- **Data securit, Monitoring and Backups:** Enhance monitoring and enable encryption on the state file and implement necessary backups for the Virtual Machine.

The technical details, timelines, and execution steps for these improvements will be captured in a dedicated Realisation document to ensure alignment with the outlined recommendations.

4. Conclusion

This security assessment highlights critical vulnerabilities in OpenRemote's Azure deployment pipeline, including over-permissioning, public IP exposure, and insufficient monitoring and backup strategies. By addressing these risks through the recommended actions, the pipeline can be significantly strengthened to align with industry best practices and the NIST Cybersecurity Framework.

Implementing the proposed measures will not only enhance the security posture but also improve the platform's resilience against potential threats. These improvements will help ensure the integrity, confidentiality, and availability of the IoT platform, safeguarding both infrastructure and user data.

Finally, security is a continuous process. It is essential to regularly review and update the pipeline's configurations, leveraging tools such as Checkov and Azure Monitor, to adapt to evolving threats and maintain a robust security posture.