# **Azure Security Remediation Guide**

**Comprehensive Fix Strategy for 434 Issues**

**Table of Contents**

1. [Pre-Remediation Strategy](https://claude.ai/chat/78d01937-09bd-4854-9194-d32fea06b75f#pre-remediation-strategy)
2. [Critical Issues (Fix First)](https://claude.ai/chat/78d01937-09bd-4854-9194-d32fea06b75f#critical-issues)
3. [High Priority Issues](https://claude.ai/chat/78d01937-09bd-4854-9194-d32fea06b75f#high-priority-issues)
4. [Medium Priority Issues](https://claude.ai/chat/78d01937-09bd-4854-9194-d32fea06b75f#medium-priority-issues)
5. [Low Priority Issues](https://claude.ai/chat/78d01937-09bd-4854-9194-d32fea06b75f#low-priority-issues)
6. [Testing & Validation](https://claude.ai/chat/78d01937-09bd-4854-9194-d32fea06b75f#testing-validation)
7. [Rollback Procedures](https://claude.ai/chat/78d01937-09bd-4854-9194-d32fea06b75f#rollback-procedures)

**Pre-Remediation Strategy**

**Phase 1: Assessment & Planning (Week 1)**

**DO NOT SKIP THIS PHASE - Critical for Production Safety**

**Step 1: Identify Business Impact**

# Create an impact assessment spreadsheet with these columns:

# 1. Issue Type

# 2. Affected Resources

# 3. Business Owner

# 4. Application/Service Dependency

# 5. Change Window

# 6. Risk Level

**Step 2: Stakeholder Communication**

* **Notify**: Application owners, DevOps teams, Security team, Management
* **Provide**: Detailed list of changes per application/service
* **Request**: Approval for change windows
* **Schedule**: Review meetings with each team

**Step 3: Create Backup & Rollback Plan**

# Backup current configurations

az network nsg list --output json > nsg-backup-$(date +%Y%m%d).json

az role assignment list --all --output json > rbac-backup-$(date +%Y%m%d).json

az storage account list --output json > storage-backup-$(date +%Y%m%d).json

az keyvault list --output json > keyvault-backup-$(date +%Y%m%d).json

**Step 4: Establish Testing Environment**

* **Create**: Mirror production environment in DEV/TEST
* **Apply**: All fixes in TEST first
* **Validate**: Application functionality
* **Document**: Results and issues

**Critical Issues (Fix First)**

**Issue 1: Unrestricted Inbound Internet Access on Multiple Ports**

**Risk**: HIGH - Direct exposure to internet attacks, ransomware, unauthorized access

**Affected Ports**: 5000-5100, 339, 14147, 990, 9895, 443

**Understanding the Impact:**

* **Port 443 (HTTPS)**: Likely needed for web applications - DO NOT BLOCK without verification
* **Ports 5000-5100**: Often used for development servers/APIs - May need restriction
* **Port 339**: Non-standard - Investigate usage first
* **Port 14147**: Non-standard - Likely unnecessary
* **Port 990**: FTPS - High risk if exposed
* **Port 9895**: Non-standard - Investigate first

**Manual Fix Process:**

**Step 1: Identify Who's Using These Ports**

# List all NSG rules with these ports

az network nsg list --query "[].{Name:name, ResourceGroup:resourceGroup}" -o table > nsg-list.txt

# For each NSG, check specific rules

az network nsg rule list --nsg-name <NSG-NAME> --resource-group <RG-NAME> \

--query "[?destinationPortRange=='443' || destinationPortRange=='5000-5100']" \

--output table

**Step 2: Document Current Access Patterns**

# Get flow logs to see who's actually connecting

az network watcher flow-log list --output table

# Review the last 30 days of connection data

# Identify legitimate source IPs

**Step 3: Contact Application Owners**

Email Template:

Subject: Action Required - Network Security Update for [Application Name]

Hi [Team Name],

We've identified that [Resource Name] has unrestricted internet access on port [X].

Current Configuration:

- Port: [X]

- Current Access: 0.0.0.0/0 (Entire Internet)

Questions:

1. Is this port actively used by your application?

2. What are the legitimate source IPs/ranges that need access?

3. When can we schedule this change? (Provide 3 time slots)

Proposed Change:

- Restrict access to specific IP ranges

- Implement Azure Application Gateway/Firewall for port 443

- Use Azure Bastion for management access

Please respond by [Date] to avoid security policy enforcement.

Best regards,

Cloud Security Team

**Step 4: Implement Fixes by Port Type**

**For Port 443 (Web Traffic) - REQUIRES CAREFUL PLANNING:**

Option A: Keep Open BUT Add Azure Application Gateway/WAF

# This is the SAFEST approach for production web apps

# 1. Create Application Gateway with WAF

az network application-gateway create \

--name prod-appgw \

--resource-group <RG-NAME> \

--location eastus \

--sku WAF\_v2 \

--capacity 2 \

--vnet-name <VNET-NAME> \

--subnet <APPGW-SUBNET> \

--public-ip-address appgw-pip \

--http-settings-cookie-based-affinity Disabled \

--frontend-port 443 \

--http-settings-port 443 \

--http-settings-protocol Https

# 2. Configure WAF policies

az network application-gateway waf-config set \

--enabled true \

--gateway-name prod-appgw \

--resource-group <RG-NAME> \

--firewall-mode Prevention \

--rule-set-type OWASP \

--rule-set-version 3.2

# 3. Update NSG to only allow traffic from App Gateway subnet

az network nsg rule update \

--nsg-name <NSG-NAME> \

--resource-group <RG-NAME> \

--name Allow-HTTPS \

--source-address-prefixes <APPGW-SUBNET-CIDR>

Option B: Restrict to Known IPs (For internal/partner apps)

# Get list of approved IPs from stakeholders

# Example: Company offices, partner VPN endpoints

az network nsg rule update \

--nsg-name <NSG-NAME> \

--resource-group <RG-NAME> \

--name Allow-HTTPS \

--source-address-prefixes 203.0.113.0/24 198.51.100.0/24 \

--priority 100 \

--access Allow \

--protocol Tcp \

--direction Inbound \

--destination-port-ranges 443

**For Ports 5000-5100 (Development/API Ports):**

# These should NEVER be exposed to internet

# Option 1: Block completely and use VPN

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG-NAME> \

--name Deny-Dev-Ports \

--priority 100 \

--access Deny \

--protocol Tcp \

--direction Inbound \

--source-address-prefixes Internet \

--destination-port-ranges 5000-5100

# Option 2: Allow only from corporate VPN subnet

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG-NAME> \

--name Allow-Dev-Ports-VPN \

--priority 100 \

--access Allow \

--protocol Tcp \

--direction Inbound \

--source-address-prefixes <VPN-SUBNET-CIDR> \

--destination-port-ranges 5000-5100

**For Non-Standard Ports (339, 14147, 990, 9895):**

# Step 1: Investigate if needed (check with teams)

# Step 2: If not needed, BLOCK immediately

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG-NAME> \

--name Deny-Unusual-Ports \

--priority 110 \

--access Deny \

--protocol '\*' \

--direction Inbound \

--source-address-prefixes Internet \

--destination-port-ranges 339 14147 990 9895

**Step 5: Phased Rollout Plan**

Week 1: Fix non-standard ports (339, 14147, 990, 9895) - Low risk

Week 2: Fix development ports (5000-5100) - Medium risk, coordinate with dev teams

Week 3: Implement App Gateway for port 443 - High risk, needs thorough testing

Week 4: Monitor and validate all changes

**Issue 2: User with Owner Permissions at Management Group Scope**

**Risk**: CRITICAL - Can modify entire Azure tenant, delete resources, change billing

**Manual Fix Process:**

**Step 1: Identify the Users**

# List all Owner assignments at Management Group level

az role assignment list \

--scope /providers/Microsoft.Management/managementGroups/<MG-ID> \

--query "[?roleDefinitionName=='Owner']" \

--output table > owner-assignments.txt

**Step 2: Assess Legitimacy**

Questions to ask:

1. Is this person in IT Leadership? (CTO, IT Director)

2. Do they actively manage infrastructure?

3. Is this a service account? (Should use Managed Identity instead)

4. Is this a contractor/external user? (HIGH RISK)

5. When was this assigned? (Check audit logs)

**Step 3: Get Executive Approval**

Email Template:

Subject: URGENT - Review of Azure Management Group Owner Permissions

Dear [Executive],

We've identified [X] users with Owner permissions at Management Group level.

This grants them ability to:

- Delete any resource in the organization

- Modify billing and subscriptions

- Change security policies

- Grant themselves additional permissions

Users identified:

[List users and their justification if known]

Recommendation:

1. Remove Owner role from all users

2. Grant Contributor role instead (prevents permission changes)

3. Use Azure PIM for just-in-time access

4. Implement break-glass accounts for emergencies

This requires C-Level approval. Can we schedule a review meeting?

Regards,

Security Team

**Step 4: Implement Least Privilege**

**Recommended Approach: Use Azure PIM (Privileged Identity Management)**

# This requires Azure AD Premium P2

# 1. Remove permanent Owner assignments

az role assignment delete \

--assignee <USER-OBJECT-ID> \

--role Owner \

--scope /providers/Microsoft.Management/managementGroups/<MG-ID>

# 2. Grant Contributor instead (for day-to-day work)

az role assignment create \

--assignee <USER-OBJECT-ID> \

--role Contributor \

--scope /providers/Microsoft.Management/managementGroups/<MG-ID>

# 3. Configure PIM for Owner role (just-in-time)

# This must be done through Azure Portal:

# Azure AD > Privileged Identity Management > Azure Resources

# - Set activation to require approval

# - Set maximum duration to 8 hours

# - Require MFA for activation

# - Require justification

**Alternative: Break-Glass Accounts**

# Create emergency accounts with Owner but MFA + Conditional Access

# 1. Create emergency admin accounts

az ad user create \

--display-name "Emergency Admin 01" \

--user-principal-name emergencyadmin01@yourdomain.com \

--password <STRONG-PASSWORD-IN-SEALED-ENVELOPE>

# 2. Assign Owner ONLY to break-glass accounts

az role assignment create \

--assignee <EMERGENCY-ADMIN-OBJECT-ID> \

--role Owner \

--scope /providers/Microsoft.Management/managementGroups/<MG-ID>

# 3. Configure alerts for break-glass account usage

az monitor action-group create \

--name break-glass-alert \

--resource-group monitoring-rg \

--short-name bgalert \

--email-receiver name=SecurityTeam email=security@company.com

# 4. Create alert rule

az monitor activity-log alert create \

--name break-glass-used \

--resource-group monitoring-rg \

--condition category=Administrative and operationName=Microsoft.Authorization/roleAssignments/write \

--action-group break-glass-alert

**Issue 3: Stale RBAC Assignments for Deleted Identities**

**Risk**: MEDIUM - Security gap, compliance violation

**Manual Fix Process:**

**Step 1: Identify Stale Assignments**

# Export all role assignments

az role assignment list --all --output json > all-assignments.json

# Find assignments with missing principals

az role assignment list --all \

--query "[?principalName==null || principalName=='']" \

--output table > stale-assignments.txt

**Step 2: Verify These Are Actually Deleted**

# Check if the object still exists in Azure AD

az ad user show --id <OBJECT-ID> 2>&1 | grep -q "does not exist" && echo "DELETED" || echo "EXISTS"

az ad group show --id <OBJECT-ID> 2>&1 | grep -q "does not exist" && echo "DELETED" || echo "EXISTS"

az ad sp show --id <OBJECT-ID> 2>&1 | grep -q "does not exist" && echo "DELETED" || echo "EXISTS"

**Step 3: Safe Removal Process**

# SAFE: These are deleted identities, removal won't break anything

# Remove stale assignment

az role assignment delete \

--assignee <OBJECT-ID> \

--scope <SCOPE>

# Log the removal

echo "$(date) - Removed stale assignment for $OBJECT\_ID at $SCOPE" >> stale-removal-log.txt

**Automated Script for Bulk Cleanup:**

#!/bin/bash

# Save as: cleanup-stale-rbac.sh

LOGFILE="stale-rbac-cleanup-$(date +%Y%m%d-%H%M%S).log"

echo "Starting stale RBAC cleanup - $(date)" | tee -a $LOGFILE

# Get all assignments

assignments=$(az role assignment list --all --query "[?principalName==null || principalName==''].{id:id, objectId:principalId, scope:scope, role:roleDefinitionName}" -o json)

# Parse and remove

echo "$assignments" | jq -c '.[]' | while read -r assignment; do

object\_id=$(echo "$assignment" | jq -r '.objectId')

scope=$(echo "$assignment" | jq -r '.scope')

role=$(echo "$assignment" | jq -r '.role')

# Verify deletion

if az ad user show --id "$object\_id" 2>&1 | grep -q "does not exist" || \

az ad group show --id "$object\_id" 2>&1 | grep -q "does not exist" || \

az ad sp show --id "$object\_id" 2>&1 | grep -q "does not exist"; then

echo "Removing stale assignment: $object\_id - $role at $scope" | tee -a $LOGFILE

az role assignment delete --assignee "$object\_id" --scope "$scope"

if [ $? -eq 0 ]; then

echo "SUCCESS: Removed $object\_id" | tee -a $LOGFILE

else

echo "FAILED: Could not remove $object\_id" | tee -a $LOGFILE

fi

fi

done

echo "Cleanup completed - $(date)" | tee -a $LOGFILE

**High Priority Issues**

**Issue 4: Subnets Without NSG Attached**

**Risk**: HIGH - No traffic filtering, lateral movement possible

**Manual Fix Process:**

**Step 1: Identify Affected Subnets**

# List all subnets without NSG

az network vnet list --output json | \

jq -r '.[] | select(.subnets[] | .networkSecurityGroup == null) |

.name + " : " + (.subnets[] | select(.networkSecurityGroup == null) | .name)'

**Step 2: Understand Subnet Purpose**

# For each subnet, check what resources are attached

az network vnet subnet show \

--vnet-name <VNET-NAME> \

--name <SUBNET-NAME> \

--resource-group <RG-NAME> \

--query "ipConfigurations[].id" -o table

**Step 3: Create Appropriate NSG**

**Template for Web Tier Subnet:**

# Create NSG

az network nsg create \

--name nsg-web-tier \

--resource-group <RG-NAME> \

--location <LOCATION>

# Allow HTTPS from internet

az network nsg rule create \

--nsg-name nsg-web-tier \

--resource-group <RG-NAME> \

--name Allow-HTTPS-Inbound \

--priority 100 \

--direction Inbound \

--access Allow \

--protocol Tcp \

--source-address-prefixes Internet \

--source-port-ranges '\*' \

--destination-address-prefixes '\*' \

--destination-port-ranges 443

# Allow HTTP from App Gateway only

az network nsg rule create \

--nsg-name nsg-web-tier \

--resource-group <RG-NAME> \

--name Allow-HTTP-From-AppGW \

--priority 110 \

--direction Inbound \

--access Allow \

--protocol Tcp \

--source-address-prefixes <APPGW-SUBNET-CIDR> \

--destination-port-ranges 80

# Deny all other inbound

az network nsg rule create \

--nsg-name nsg-web-tier \

--resource-group <RG-NAME> \

--name Deny-All-Inbound \

--priority 4096 \

--direction Inbound \

--access Deny \

--protocol '\*' \

--source-address-prefixes '\*' \

--destination-port-ranges '\*'

**Template for App Tier Subnet:**

az network nsg create \

--name nsg-app-tier \

--resource-group <RG-NAME>

# Allow from web tier only

az network nsg rule create \

--nsg-name nsg-app-tier \

--resource-group <RG-NAME> \

--name Allow-From-Web-Tier \

--priority 100 \

--direction Inbound \

--access Allow \

--protocol Tcp \

--source-address-prefixes <WEB-SUBNET-CIDR> \

--destination-port-ranges 8080 8443

# Deny all other

az network nsg rule create \

--nsg-name nsg-app-tier \

--resource-group <RG-NAME> \

--name Deny-All-Inbound \

--priority 4096 \

--direction Inbound \

--access Deny \

--protocol '\*' \

--source-address-prefixes '\*' \

--destination-port-ranges '\*'

**Step 4: Attach NSG to Subnet (CAREFUL - Test First)**

# IMPORTANT: This can break connectivity if rules are wrong

# Test in non-prod first!

az network vnet subnet update \

--vnet-name <VNET-NAME> \

--name <SUBNET-NAME> \

--resource-group <RG-NAME> \

--network-security-group nsg-web-tier

# Monitor for 1 hour

# Check application logs

# Verify connectivity

**Step 5: Gradual Rollout**

1. Create NSG with allow-all rules initially

2. Attach to subnet

3. Monitor for 24 hours

4. Gradually restrict rules

5. Monitor after each change

**Issue 5: Storage Account Security Issues**

**Sub-Issue 5A: Storage Account Allows Public Access**

**Risk**: HIGH - Data breach, unauthorized access

**Step 1: Identify Public Storage Accounts**

az storage account list --query "[?allowBlobPublicAccess==true || publicNetworkAccess=='Enabled']" -o table

**Step 2: Check if Public Access is Required**

# Check usage patterns

az monitor activity-log list \

--resource-id /subscriptions/<SUB-ID>/resourceGroups/<RG>/providers/Microsoft.Storage/storageAccounts/<SA-NAME> \

--start-time 2024-01-01T00:00:00Z \

--query "[?contains(operationName.value, 'blob')]" -o table

# Identify if any containers need public access

az storage container list \

--account-name <SA-NAME> \

--query "[?properties.publicAccess != 'None']" -o table

**Step 3: Implement Fix Based on Use Case**

**Case 1: No Public Access Needed (Most Common)**

# Disable public blob access

az storage account update \

--name <SA-NAME> \

--resource-group <RG-NAME> \

--allow-blob-public-access false

# Restrict network access

az storage account update \

--name <SA-NAME> \

--resource-group <RG-NAME> \

--default-action Deny

# Add allowed networks

az storage account network-rule add \

--account-name <SA-NAME> \

--resource-group <RG-NAME> \

--vnet-name <VNET-NAME> \

--subnet <SUBNET-NAME>

**Case 2: Public Access Required for Static Website**

# Keep public access BUT use CDN with security features

az cdn profile create \

--name cdn-profile \

--resource-group <RG-NAME> \

--sku Standard\_Microsoft

az cdn endpoint create \

--name cdn-endpoint \

--profile-name cdn-profile \

--resource-group <RG-NAME> \

--origin <SA-NAME>.blob.core.windows.net \

--origin-host-header <SA-NAME>.blob.core.windows.net \

--enable-compression true

# Configure WAF on CDN

az cdn waf policy create \

--name cdn-waf-policy \

--resource-group <RG-NAME> \

--sku Standard\_Microsoft

# Then restrict storage account to CDN IPs only

**Case 3: Temporary Public Access Required**

# Use SAS tokens instead of public access

az storage container generate-sas \

--account-name <SA-NAME> \

--name <CONTAINER-NAME> \

--permissions r \

--expiry 2025-12-31T23:59:59Z \

--output tsv

# Share SAS URL instead of making container public

**Sub-Issue 5B: Storage Account Not Enforcing TLS 1.2**

**Risk**: MEDIUM - Vulnerable to protocol downgrade attacks

**Fix (SAFE - No impact on modern clients):**

# Set minimum TLS version to 1.2

az storage account update \

--name <SA-NAME> \

--resource-group <RG-NAME> \

--min-tls-version TLS1\_2

# Also enforce HTTPS only

az storage account update \

--name <SA-NAME> \

--resource-group <RG-NAME> \

--https-only true

**Verification:**

# Test connection with old TLS

curl --tlsv1.1 https://<SA-NAME>.blob.core.windows.net

# Should fail

# Test with TLS 1.2

curl --tlsv1.2 https://<SA-NAME>.blob.core.windows.net

# Should work

**Issue 6: Key Vault Without Purge Protection**

**Risk**: MEDIUM - Accidental deletion, compliance violation

**Step 1: Check Current Status**

az keyvault list --query "[?properties.enablePurgeProtection!=true]" -o table

**Step 2: Enable Soft Delete and Purge Protection**

# Enable soft delete (if not already)

az keyvault update \

--name <KV-NAME> \

--enable-soft-delete true \

--retention-days 90

# Enable purge protection (CANNOT BE DISABLED ONCE ENABLED)

az keyvault update \

--name <KV-NAME> \

--enable-purge-protection true

**WARNING**: Purge protection CANNOT be disabled once enabled. This is permanent.

**Best Practice:**

# Before enabling purge protection, ensure:

# 1. You have a backup process

# 2. You understand you cannot force-delete the Key Vault for 90 days after deletion

# 3. Management approves this change

# Document this decision

echo "$(date) - Enabled purge protection on <KV-NAME> - Approved by <NAME>" >> keyvault-changes.log

**Issue 7: NSG Missing Explicit Default Deny Rule**

**Risk**: MEDIUM - Relying on implicit deny

**Fix (SAFE - No impact as Azure has implicit deny):**

# Add explicit deny all rule at lowest priority

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG-NAME> \

--name Deny-All-Inbound \

--priority 4096 \

--direction Inbound \

--access Deny \

--protocol '\*' \

--source-address-prefixes '\*' \

--source-port-ranges '\*' \

--destination-address-prefixes '\*' \

--destination-port-ranges '\*'

# Also add for outbound (if needed)

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG-NAME> \

--name Deny-All-Outbound \

--priority 4096 \

--direction Outbound \

--access Deny \

--protocol '\*' \

--source-address-prefixes '\*' \

--destination-port-ranges '\*'

**Testing & Validation**

**Pre-Implementation Testing Checklist**

[ ] Backup all current configurations

[ ] Document current state

[ ] Identify critical applications/services

[ ] Schedule maintenance windows

[ ] Notify stakeholders

[ ] Prepare rollback procedures

[ ] Test in non-production environment

[ ] Create monitoring alerts

[ ] Brief on-call team

[ ] Prepare incident response plan

**Post-Implementation Validation**

# Test network connectivity

az network watcher connection-monitor create \

--name connectivity-monitor \

--resource-group <RG-NAME> \

--location <LOCATION>

# Monitor for errors

az monitor activity-log list \

--start-time $(date -u -d '1 hour ago' '+%Y-%m-%dT%H:%M:%SZ') \

--query "[?level=='Error']" -o table

# Check application health

# (Use your application-specific health checks)

**Rollback Procedures**

**Network Changes Rollback**

# Restore NSG from backup

az network nsg delete --name <NSG-NAME> --resource-group <RG-NAME>

az network nsg create --name <NSG-NAME> --resource-group <RG-NAME>

# Restore rules from backup JSON

cat nsg-backup-YYYYMMDD.json | jq -r '.[] |

"az network nsg rule create --nsg-name \(.name) --resource-group \(.resourceGroup) ..."'

**RBAC Changes Rollback**

# Restore role assignments from backup

cat rbac-backup-YYYYMMDD.json | jq -c '.[]' | while read assignment; do

az role assignment create \

--assignee $(echo $assignment | jq -r '.principalId') \

--role $(echo $assignment | jq -r '.roleDefinitionName') \

--scope $(echo $assignment | jq -r '.scope')

done

**Summary Recommendations**

**Immediate Actions (This Week)**

1. Fix stale RBAC assignments (No risk)
2. Enable TLS 1.2 on storage accounts (Low risk)
3. Block unusual ports (339, 14147, 990, 9895)
4. Enable Key Vault purge protection

**Short-Term Actions (Next 2-4 Weeks)**

1. Review Owner permissions - get executive approval
2. Implement NSGs on subnets (with careful testing)
3. Restrict development ports (5000-5100)
4. Disable public storage access where not needed

**Long-Term Actions (Next 1-3 Months)**

1. Implement Azure Application Gateway with WAF
2. Deploy Azure Bastion for secure VM access
3. Implement Azure PIM for privileged access
4. Set up comprehensive monitoring and alerting
5. Establish regular security review process

**Communication Template for Client**

Subject: Azure Security Remediation Plan - 434 Issues Identified

Executive Summary:

Our security scan identified 434 configuration issues in your Azure environment.

These range from critical security gaps to minor compliance items.

Risk Breakdown:

- Critical: [X] issues (Management access, public exposure)

- High: [X] issues (Network security, permissions)

- Medium: [X] issues (Encryption, compliance)

- Low: [X] issues (Governance, tagging)

Proposed Remediation Timeline:

- Phase 1 (Week 1-2): Low-risk fixes, no application impact

- Phase 2 (Week 3-4): Network security with coordination

- Phase 3 (Week 5-8): Comprehensive security improvements

Required From Your Team:

1. Application owner contacts for each service

2. Approved maintenance windows

3. Executive approval for permission changes

4. Budget approval for security tools (App Gateway, PIM)

Next Steps:

1. Review detailed remediation plan (attached)

2. Schedule kickoff meeting

3. Assign internal stakeholders

4. Begin phase 1 fixes

Please respond with approval to proceed.