**Azure Security Remediation - Quick Reference Checklist**

**434 Issues - Phased Rollout Plan**

**Executive Summary**

* **Total Issues**: 434
* **Estimated Timeline**: 6-8 weeks
* **Risk Level**: Manageable with proper planning
* **Budget Impact**: Minimal (mostly configuration changes)

**Phase 1: Zero-Risk Fixes (Week 1)**

**Can be done immediately - No application impact**

**1.1 Stale RBAC Assignments**

* [ ] **Task**: Remove deleted identity permissions
* [ ] **Risk**: NONE (identities already deleted)
* [ ] **Time**: 2 hours
* [ ] **Command**:

.\Safe-Remediation.ps1 -Environment PROD -FixCategories StaleRBAC -DryRun

# Review output

.\Safe-Remediation.ps1 -Environment PROD -FixCategories StaleRBAC

**1.2 Storage Account TLS Enforcement**

* [ ] **Task**: Enforce TLS 1.2 on all storage accounts
* [ ] **Risk**: NONE (all modern clients support TLS 1.2)
* [ ] **Time**: 1 hour
* [ ] **Command**:

# Test one storage account first

az storage account update --name <SA-NAME> --resource-group <RG> --min-tls-version TLS1\_2

# If successful, run automation

.\Safe-Remediation.ps1 -Environment PROD -FixCategories StorageSecurity

**1.3 Storage Account HTTPS Only**

* [ ] **Task**: Disable HTTP access
* [ ] **Risk**: NONE (HTTP should never be used)
* [ ] **Time**: 1 hour
* [ ] **Included**: In StorageSecurity fix above

**1.4 NSG Explicit Deny Rules**

* [ ] **Task**: Add explicit deny all at priority 4096
* [ ] **Risk**: NONE (Azure already has implicit deny)
* [ ] **Time**: 2 hours
* [ ] **Command**:

.\Safe-Remediation.ps1 -Environment PROD -FixCategories NSGDeny -DryRun

# Review then execute

.\Safe-Remediation.ps1 -Environment PROD -FixCategories NSGDeny

**Phase 1 Deliverable**: ~50-100 issues resolved with zero risk

**Phase 2: Low-Risk Fixes (Week 2-3)**

**Requires coordination but low application impact**

**2.1 Key Vault Purge Protection**

* [ ] **Task**: Enable soft delete and purge protection
* [ ] **Risk**: LOW (but permanent - cannot be undone)
* [ ] **Time**: 2 hours
* [ ] **Pre-requisite**: Get management approval (permanent change)
* [ ] **Command**:

# Enable on each Key Vault after approval

az keyvault update --name <KV-NAME> --enable-soft-delete true --retention-days 90

az keyvault update --name <KV-NAME> --enable-purge-protection true

**2.2 Block Non-Standard Ports**

* [ ] **Task**: Block ports 339, 14147, 990, 9895
* [ ] **Risk**: LOW (these are unusual ports)
* [ ] **Pre-requisite**: Confirm with dev teams these aren't used
* [ ] **Time**: 3 hours
* [ ] **Process**:

# Step 1: Check if ports are actually being used

az monitor activity-log list --max-events 1000 | grep -E "339|14147|990|9895"

# Step 2: Email dev teams for confirmation

# Step 3: Block ports

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG> \

--name Deny-Unusual-Ports \

--priority 110 \

--access Deny \

--protocol '\*' \

--direction Inbound \

--source-address-prefixes Internet \

--destination-port-ranges 339 14147 990 9895

**2.3 Restrict Development Ports (5000-5100)**

* [ ] **Task**: Block or restrict to VPN only
* [ ] **Risk**: MEDIUM (may break dev access)
* [ ] **Pre-requisite**: Get dev team approval and VPN subnet
* [ ] **Time**: 4 hours
* [ ] **Process**:

# Option 1: Block completely (recommended)

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG> \

--name Deny-Dev-Ports \

--priority 100 \

--access Deny \

--protocol Tcp \

--source-address-prefixes Internet \

--destination-port-ranges 5000-5100

# Option 2: Allow from VPN only

az network nsg rule create \

--nsg-name <NSG-NAME> \

--resource-group <RG> \

--name Allow-Dev-Ports-VPN \

--priority 100 \

--access Allow \

--protocol Tcp \

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

--destination-port-ranges 5000-5100

**Phase 2 Deliverable**: ~100-150 additional issues resolved

**Phase 3: Medium-Risk Fixes (Week 4-5)**

**Requires careful testing and coordination**

**3.1 Storage Account Public Access**

* [ ] **Task**: Disable public blob access
* [ ] **Risk**: HIGH (may break applications)
* [ ] **Pre-requisite**: Identify which apps use public access
* [ ] **Time**: 8-12 hours
* [ ] **Process**:

**Step 1: Identify Public Storage Usage (Week 4)**

# Find storage accounts with public access

az storage account list --query "[?allowBlobPublicAccess==true]" -o table

# For each account, check containers

az storage container list --account-name <SA-NAME> --query "[?properties.publicAccess != 'None']"

# Check access logs for external access

az monitor activity-log list --resource-id <STORAGE-RESOURCE-ID> --start-time <30-DAYS-AGO>

**Step 2: Contact Application Owners**

* [ ] Send email to each application owner
* [ ] Request alternative authentication methods
* [ ] Provide SAS token solution examples
* [ ] Get approval for change window

**Step 3: Implement Changes (Week 5)**

# For each storage account confirmed:

# Method 1: Disable public access (preferred)

az storage account update \

--name <SA-NAME> \

--resource-group <RG> \

--allow-blob-public-access false

# Method 2: Use SAS tokens for temporary access

az storage container generate-sas \

--account-name <SA-NAME> \

--name <CONTAINER> \

--permissions r \

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

--output tsv

# Method 3: Use Private Endpoints for internal access

az network private-endpoint create \

--name pe-storage \

--resource-group <RG> \

--vnet-name <VNET> \

--subnet <SUBNET> \

--private-connection-resource-id <STORAGE-RESOURCE-ID> \

--group-id blob \

--connection-name pe-connection

**3.2 Attach NSGs to Subnets**

* [ ] **Task**: Create and attach NSGs to unprotected subnets
* [ ] **Risk**: HIGH (can break connectivity)
* [ ] **Pre-requisite**: Map all subnet traffic patterns
* [ ] **Time**: 12-16 hours
* [ ] **Process**:

**Step 1: Inventory Subnets (Week 4)**

# Find subnets without NSGs

az network vnet list --output json | \

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

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

# For each subnet, identify resources

az network vnet subnet show \

--vnet-name <VNET> \

--name <SUBNET> \

--resource-group <RG> \

--query "ipConfigurations[].id"

**Step 2: Design NSG Rules (Week 4)**

* [ ] Document required inbound traffic
* [ ] Document required outbound traffic
* [ ] Create NSG rule matrix
* [ ] Get application owner approval

**Step 3: Create NSG with Allow-All Initially (Week 5)**

# Create NSG with permissive rules first

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

# Allow all inbound initially (temporary)

az network nsg rule create \

--nsg-name nsg-<SUBNET-NAME> \

--resource-group <RG> \

--name Temp-Allow-All \

--priority 100 \

--access Allow \

--protocol '\*' \

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

--destination-port-ranges '\*'

# Attach to subnet

az network vnet subnet update \

--vnet-name <VNET> \

--name <SUBNET> \

--resource-group <RG> \

--network-security-group nsg-<SUBNET-NAME>

# Monitor for 24 hours

**Step 4: Gradually Restrict Rules (Week 5)**

# Day 1: Add specific allow rules

az network nsg rule create --nsg-name nsg-<SUBNET-NAME> --name Allow-HTTPS --priority 110 ...

# Day 2: Lower priority of Temp-Allow-All

az network nsg rule update --nsg-name nsg-<SUBNET-NAME> --name Temp-Allow-All --priority 4000

# Day 3: Add deny rule

az network nsg rule create --nsg-name nsg-<SUBNET-NAME> --name Deny-All --priority 4096 --access Deny

# Day 4: Remove Temp-Allow-All

az network nsg rule delete --nsg-name nsg-<SUBNET-NAME> --name Temp-Allow-All

**Phase 3 Deliverable**: ~150-200 additional issues resolved

**Phase 4: High-Risk Fixes (Week 6-8)**

**Requires extensive planning and executive approval**

**4.1 Review Management Group Owner Permissions**

* [ ] **Task**: Remove unnecessary Owner roles
* [ ] **Risk**: CRITICAL (can lock out admins)
* [ ] **Pre-requisite**: Executive approval required
* [ ] **Time**: 8-12 hours
* [ ] **Process**:

**Step 1: Identify and Document (Week 6)**

# 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-report.txt

# Create spreadsheet with:

# - User name

# - Email

# - Department

# - Justification for Owner role

# - Last activity date

# - Proposed new role

**Step 2: Executive Review Meeting (Week 6)**

* [ ] Schedule C-level meeting
* [ ] Present findings
* [ ] Propose PIM (Privileged Identity Management)
* [ ] Get written approval

**Step 3: Implement Changes (Week 7)**

# Implement Azure PIM (requires Azure AD Premium P2)

# This is done through Portal:

# Azure AD > Privileged Identity Management > Azure Resources

# Remove permanent Owner, add Contributor

az role assignment delete \

--assignee <USER-OBJECT-ID> \

--role Owner \

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

az role assignment create \

--assignee <USER-OBJECT-ID> \

--role Contributor \

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

# Configure PIM for just-in-time Owner access

# - Require approval

# - Max duration: 8 hours

# - Require MFA

# - Require justification

**4.2 Restrict Port 443 (HTTPS)**

* [ ] **Task**: Implement Azure Application Gateway with WAF
* [ ] **Risk**: CRITICAL (can break production websites)
* [ ] **Pre-requisite**: Budget approval, extensive testing
* [ ] **Time**: 20-30 hours
* [ ] **Budget**: ~$200-500/month for App Gateway
* [ ] **Process**:

**Step 1: Design (Week 6)**

* [ ] Identify all web applications on port 443
* [ ] Map backend pools
* [ ] Design health probes
* [ ] Plan SSL certificates
* [ ] Design routing rules

**Step 2: Deploy in Parallel (Week 7)**

# Create Application Gateway

az network application-gateway create \

--name prod-appgw \

--resource-group <RG> \

--location eastus \

--sku WAF\_v2 \

--capacity 2 \

--vnet-name <VNET> \

--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

# Configure WAF policies

az network application-gateway waf-config set \

--enabled true \

--gateway-name prod-appgw \

--resource-group <RG> \

--firewall-mode Prevention \

--rule-set-type OWASP \

--rule-set-version 3.2

# Add backend pools for each application

az network application-gateway address-pool create \

--gateway-name prod-appgw \

--resource-group <RG> \

--name backend-app1 \

--servers <BACKEND-IP>

**Step 3: Test (Week 7)**

* [ ] Point test domain to App Gateway
* [ ] Verify functionality
* [ ] Load test
* [ ] Security scan
* [ ] Get user acceptance

**Step 4: Cutover (Week 8)**

* [ ] Schedule maintenance window (weekend)
* [ ] Update DNS to point to App Gateway
* [ ] Monitor for 24 hours
* [ ] Update NSG to allow only from App Gateway subnet

az network nsg rule update \

--nsg-name <NSG> \

--resource-group <RG> \

--name Allow-HTTPS \

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

**Phase 4 Deliverable**: Remaining critical issues resolved

**Daily Operations Checklist**

**Before Making Any Changes**

* [ ] Verify current Azure context: az account show
* [ ] Create configuration backup
* [ ] Notify on-call team
* [ ] Check for ongoing incidents
* [ ] Verify rollback procedure

**After Making Changes**

* [ ] Test application functionality
* [ ] Check Azure Activity Log for errors
* [ ] Monitor for 1 hour
* [ ] Document changes
* [ ] Update runbooks

**Rollback Triggers**

Execute rollback if:

* [ ] Application error rate increases >5%
* [ ] Response time increases >50%
* [ ] User reports of access issues
* [ ] Critical alerts fired
* [ ] Security scan fails

**Communication Templates**

**Pre-Change Notification**

Subject: Scheduled Azure Security Update - [Date/Time]

Team,

We will be performing security updates on [Date] at [Time].

Changes:

- [List specific changes]

Expected Impact:

- [None/Brief/Extended]

Services Affected:

- [List services]

Rollback Plan:

- [Describe rollback]

Point of Contact:

- Primary: [Name/Phone]

- Backup: [Name/Phone]

Please respond if you have concerns.

**Post-Change Report**

Subject: Completed - Azure Security Update

Team,

Security updates completed successfully.

Changes Applied:

- [List changes]

Results:

- [Success metrics]

Issues Encountered:

- [None/List issues and resolutions]

Next Actions:

- [List any follow-up required]

Monitoring:

- Continuing to monitor for [duration]

**Emergency Rollback Procedures**

**NSG Changes Rollback**

# Restore from backup

backup\_file="nsg-backup-YYYYMMDD.json"

# Delete current NSG

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

# Recreate from backup (manual process - review backup file)

**RBAC Changes Rollback**

# Restore specific assignment

az role assignment create \

--assignee <OBJECT-ID> \

--role <ROLE-NAME> \

--scope <SCOPE>

**Storage Account Rollback**

# Re-enable public access (if needed)

az storage account update \

--name <SA-NAME> \

--resource-group <RG> \

--allow-blob-public-access true

**Success Metrics**

**Week 1-2 Targets**

* [ ] 50+ issues resolved
* [ ] Zero application incidents
* [ ] All backups created
* [ ] Stakeholder approval obtained

**Week 3-4 Targets**

* [ ] 150+ issues resolved
* [ ] Network security improved
* [ ] Port restrictions implemented
* [ ] NSG deny rules in place

**Week 5-6 Targets**

* [ ] 300+ issues resolved
* [ ] Storage security hardened
* [ ] Subnet NSGs deployed
* [ ] Public access eliminated

**Week 7-8 Targets**

* [ ] All 434 issues resolved
* [ ] App Gateway deployed
* [ ] PIM implemented
* [ ] Zero critical findings

**Budget Requirements**

**One-Time Costs**

* **Azure Application Gateway**: $0 (included in Azure)
* **Consulting Hours**: 40-60 hours @ $150/hr = $6,000-9,000

**Recurring Costs**

* **Application Gateway**: ~$250/month
* **Azure AD Premium P2** (for PIM): ~$9/user/month
* **Storage egress** (may increase with private endpoints): Variable

**Total Estimated Budget**

* **One-time**: $6,000-9,000
* **Monthly**: $300-500

**Final Recommendations**

**Do These First (Safe)**

1. Stale RBAC cleanup
2. Enable TLS 1.2
3. Enable HTTPS only
4. Add NSG deny rules
5. Enable Key Vault protections

**Do These with Caution**

1. Disable storage public access
2. Attach NSGs to subnets
3. Block development ports
4. Restrict unusual ports

**Do These with Executive Approval**

1. Remove Owner permissions
2. Implement PIM
3. Deploy Application Gateway
4. Restructure network access

**Never Do These**

1. Make changes without backup
2. Skip testing in non-prod
3. Apply all fixes at once
4. Make changes during business hours (for high-risk items)
5. Proceed without stakeholder approval

**Questions for Client**

Before beginning, please answer:

1. **Do you have a test/dev Azure subscription for testing?**
   * [ ] Yes - Mirror of production
   * [ ] Yes - Partial mirror
   * [ ] No - Need to create
2. **What are your maintenance windows?**
   * [ ] Weekends: [Time]
   * [ ] Weekdays after hours: [Time]
   * [ ] By appointment only
3. **Who are the application owners for affected resources?**
   * [ ] Have complete list
   * [ ] Need to identify
   * [ ] Single team manages all
4. **Do you have Azure AD Premium P2?**
   * [ ] Yes
   * [ ] No - Need to purchase for PIM
5. **What is your approval process for production changes?**
   * [ ] CAB meeting required
   * [ ] Director approval
   * [ ] Email approval sufficient
6. **Current monitoring tools?**
   * [ ] Azure Monitor
   * [ ] Application Insights
   * [ ] Third-party (specify)
   * [ ] None - need to implement

**Support Contact**

For questions or assistance during remediation:

* **Email**: security-team@company.com
* **On-call**: [Phone number]
* **Escalation**: [Manager name/contact]