Azure-IAM-Hardening

This project is Part B of my IAM Hardening Lab series — focused on Microsoft Azure. It mirrors enterprise-grade identity governance by enforcing multi-factor authentication (MFA), leveraging Privileged Identity Management (PIM), and validating compliance with CIS benchmarks using Prowler.

Project Goals

- Create overprivileged and least-privileged Azure test users
- Configure Microsoft Entra P2 licenses and PIM
- Enable and validate Conditional Access with MFA
- Run Prowler for CIS 3.0 benchmark scans via Docker

Folder Structure

```
iam-hardening-project/
│
├── demo-policies/            # Sample JSON policies and summaries
│   └── conditional-access-summary.md
│
├── evidence/                 # Evidence of test and audit completion
│   └── prowler_cis_3.0_azure_report.csv
│
├── scripts/                  # Optional helper scripts
│   └── create-service-principal.sh (optional placeholder)
│
└── README.md
```

Identity Hardening Project (Azure - Part B)

1. Create Azure AD Users

- **stan_overpriv** → Over-privileged, admin-level account
- **chris_leastpriv** → Restricted, least-privilege account

Tasks completed:

- Configured `usageLocation` to avoid license assignment errors
- Assigned Microsoft Entra ID Premium P2 licenses to support PIM
- Validated role visibility and access scope in the Azure Portal

2. Enable Privileged Identity Management (PIM)

- Enabled and configured PIM to enforce just-in-time access controls
- Assigned **eligible** (not permanent) roles
- Required MFA for all role activations
- Limited activation duration to **1 hour**
- Required justification (ticket ID + reason)
- Validated activation flow using `stan_overpriv` for Owner role

Key takeaway: PIM enforces least privilege, auditability, and time-based controls.

3. Configure Default Security Settings & Conditional Access

- Ensured Conditional Access (CA) was configured before enabling default settings

Sequence followed:

1. License assignment
2. PIM configuration and testing
3. Conditional Access setup (Require MFA for all users)
4. Enable Default Security Settings (last)

4. Compliance Scanning with Prowler (CIS 3.0 Azure)

4.1 Install Docker Desktop + WSL Integration

- Installed Docker Desktop with Ubuntu via WSL2

```
wsl --install
```

4.2 Authenticate Azure CLI

```
az login
```

4.3 Create Service Principal for Prowler Scan

```
az ad sp create-for-rbac
  --name prowler-sp
  --role Reader
  --scopes /subscriptions/<SUBSCRIPTION_ID>
```

This returns:

```
appId → Client ID
password → Client Secret
tenant → Tenant ID
```

4.4 Export Azure Credentials in Ubuntu

```
export AZURE_CLIENT_ID=<YOUR_CLIENT_ID>
export AZURE_CLIENT_SECRET=<YOUR_CLIENT_SECRET>
export AZURE_TENANT_ID=<YOUR_TENANT_ID>
export AZURE_SUBSCRIPTION_ID=<YOUR_SUBSCRIPTION_ID>
```

4.5 Run Prowler via Docker

```
sudo docker run --rm -it
  -v "$(pwd)/output:/output"
  -e AZURE_CLIENT_ID=$AZURE_CLIENT_ID
  -e AZURE_CLIENT_SECRET=$AZURE_CLIENT_SECRET
  -e AZURE_TENANT_ID=$AZURE_TENANT_ID
  -e AZURE_SUBSCRIPTION_ID=$AZURE_SUBSCRIPTION_ID
  toniblyx/prowler:latest
  azure
  --subscription-id $AZURE_SUBSCRIPTION_ID
  --sp-env-auth
  --compliance cis_3.0_azure
  -M csv
  -o /output
```

Output saved to:

```
evidence/prowler_cis_3.0_azure_report.csv
```

4.6 Example Compliance Output

```
Compliance Status of CIS_3.0_AZURE Framework:
    100% FAIL (17) | 0% PASS (0) | 0% MUTED (0)

Framework Breakdown:
Provider   Section                L1      L2
Azure      Security              PASS(0) FAIL(2)
Azure      Logging & Monitoring  FAIL(12) FAIL(1)
Azure      Networking            PASS(0) FAIL(1)
Azure      Virtual Machines      PASS(0) FAIL(1)
```

Summary

This project showcased hands-on IAM governance and audit simulation in Azure:

- Enforced MFA and Conditional Access
- Leveraged PIM for just-in-time role elevation
- Created overprivileged vs. least-privileged accounts
- Validated compliance posture using Prowler

Author

Kent Ward
GRC Analyst → GRC Engineer (in progress)
Cloud Security | IAM | Audit | Compliance
GitHub: github.com/Kent-Ward
LinkedIn: linkedin.com/in/kent-ward-75b440265

Notes

> All secrets and IDs have been redacted for GitHub sharing.
>
> Replace `<SUBSCRIPTION_ID>` and credential variables with your own when replicating.
>
> Docker runs as root, so mount folders from `/home/your-username/` or fix permissions using `chown`.
>
> Folder structure and evidence files mirror enterprise audit practices.