THOMAS MORE

Lab 5.1

Compliance Violations in the Current Deployment

Toepasbare Cloud Security voor IT-Professionals

2025 - 2026

Contents

1. Compliance Violations in the Current Deployment

1.1. Context

The Day 5 environment contains the main cloud resources from the course application:

- App Services.
- Key Vault.
- Storage accounts.
- Log Analytics workspace.
- Application Insights.
- Virtual network and network security resources.

In this lab, you will manually audit the deployment for governance and security findings.

You are not fixing anything yet. The goal is to find, understand, and document issues that would matter in a real security or compliance review.

Typical findings may include:

- Missing or inconsistent tags.
- Storage accounts that are reachable from public networks.
- Storage accounts where anonymous blob access needs review.
- Missing diagnostic settings for security-sensitive resources.
- Development settings that would not be acceptable in production.
- Network rules that are too broad or too restrictive.

> **Note**
>
> The Day 5 infrastructure is intentionally close to the earlier course environment, but with settings that make sense for a lab or development deployment. Some of those settings would be findings in a production governance review.

1.2. Learning goals

By the end of this lab, you should be able to explain:

- What compliance and governance findings exist in the current deployment.
- Why required tags matter for ownership, cost, automation, and incident response.
- The difference between storage public network access and anonymous blob access.
- Why Key Vault diagnostic settings are important for audit trails.
- How development settings can become a production risk.
- Why manual audits do not scale well.

1.3. Estimated time

45 minutes

Suggested timing:

Time	Activity
10 minutes	Review the Terraform environment configuration
15 minutes	Audit resources in the Azure portal
10 minutes	Document findings in a compliance checklist
5 minutes	Classify findings by severity
5 minutes	Reflection and discussion

1.4. Before you start

Make sure you have:

- [ ] Day 5 infrastructure deployed.
- [ ] Access to the correct Azure tenant and subscription.
- [ ] Reader access or higher on the resource group.
- [ ] Access to the Azure portal.
- [ ] The Day 5 resource group name.

> **Note**
>
> Reader access is enough for most audit steps. Contributor is only needed when you start fixing findings in the next lab.

2. Confirm Your Azure Context

Before auditing resources, confirm that you are looking at the correct environment.

Run:

```
az account show --output table
```

Write down:

Item	Value
Subscription name	
Subscription ID	
Tenant ID	
Signed-in user	

> **Question**
> Why is it important to confirm the active subscription before starting a compliance audit?

3. Review the Terraform Configuration

Open the Day 5 Terraform configuration.

The exact path may differ in your repository, but it will usually be in a Day 5 infrastructure folder.

Look for the environment configuration block, for example:

```
env_config = {
  dev = {
    sku                        = "P0v3"
    log_retention              = 30
    enable_private_endpoints   = false
    enable_diagnostic_logs     = true
    require_https              = true
    public_network_access      = "Enabled"
    infrastructure_encryption  = false
  }
  prod = {
    sku                        = "P0v3"
    log_retention              = 90
    enable_private_endpoints   = true
    enable_diagnostic_logs     = true
    require_https              = true
    public_network_access      = "Disabled"
    infrastructure_encryption  = true
  }
}
```

> **Note**
>
> Treat this block as the intended design, not as proof of the actual deployed state. A real audit compares what the code says, what the cloud resources actually do, and what the security baseline requires.

3.1. Development versus production

The example configuration shows a common difference:

Setting	Dev example	Prod example
Public network access	`Enabled`	`Disabled`
Private endpoints	`false`	`true`
Log retention	`30` days	`90` days
Infrastructure encryption	`false`	`true`

> **Note**
>
> Azure Storage is encrypted at rest by default. Infrastructure encryption is an additional double-encryption option for supported storage accounts. It must be enabled when the storage account is created and cannot be toggled later on the same account.

Question

Why might a development environment allow more public access than production? What risk appears when development settings are copied into production?

Question

Why is infrastructure encryption usually a design-time decision rather than a quick manual fix?

4. Discover the Resources

List the resources in your Day 5 resource group.

```
az resource list `
  --resource-group <RESOURCE_GROUP_NAME> `
  --output table
```

> **Note**
>
> The examples use PowerShell-style line continuation with a backtick. If you use Bash, replace the backtick with a backslash.

Write down the main resources:

Resource name	Resource type	Purpose

5. Audit Resource Tags

Tags help identify ownership, cost allocation, lifecycle, automation scope, and incident response context.

Expected baseline tags:

Tag	Expected value
`environment`	`dev` or the environment selected for the lab
`student`	`<your_prefix>`
`managed_by`	`terraform`
`day`	`day-5`

5.1. Check tags in the portal

For each important resource:

1. Open the resource.
2. Open **Tags**.
3. Record missing or unexpected tags.

5.2. Optional CLI check

```
az resource list `
  --resource-group <RESOURCE_GROUP_NAME> `
  --query "[].{Name:name, Type:type, Tags:tags}" `
  --output table
```

5.3. Tag audit table

Resource	Expected tags	What you find
Public backend App Service	`environment`, `student`, `managed_by`, `day`	
Internal backend App Service	`environment`, `student`, `managed_by`, `day`	
Key Vault	`environment`, `student`, `managed_by`, `day`	
Frontend storage account	`environment`, `student`, `managed_by`, `day`	
Private/data storage account	`environment`, `student`, `managed_by`, `day`	
Log Analytics workspace	`environment`, `student`, `managed_by`, `day`	
Application Insights	`environment`, `student`, `managed_by`, `day`	

Note

Not every Azure child resource behaves the same way for tags. For this lab, focus on the main resources in your resource group.

Question

Which tag would be most useful during an incident: `owner`, `environment`, `managed_by`, or `day`? Why?

6. Audit Storage Exposure

Storage exposure has multiple layers. Do not mark everything as simply “public” without checking what kind of access is allowed.

Control	What it means
Public network access	Whether the storage account endpoint is reachable over public networks.
Firewall rules	Which public IP ranges, virtual networks, or trusted Azure services may reach the storage account.
Private endpoint	Whether the storage account has private connectivity through a virtual network.
Allow Blob anonymous access	Whether blob containers may be configured for unauthenticated read access.
Container access level	Whether a specific container allows anonymous access, if account-level anonymous access permits it.

> **Note**
>
> A storage account can be reachable over a public endpoint and still require authentication. That is different from allowing anonymous blob reads.

6.1. Frontend storage account

The frontend storage account may be used for static website hosting.

Check:

- Is static website hosting enabled?
- Is public network access enabled?
- Is anonymous blob access enabled or disabled?
- Are containers configured for anonymous access?

> **Note**
>
> Frontend hosting may intentionally require public reachability. That does not mean all storage accounts should be public. Classify the finding based on the purpose of the account.

6.2. Private or data storage account

For a storage account used for private application data, check:

- Is public network access enabled?
- Are firewall rules configured?
- Is there a private endpoint?
- Is anonymous blob access disabled?

6.3. Storage audit table

Resource	Public network access	Anonymous blob access	Finding
Frontend storage account			
Private/data storage account			

> **Question**
>
> Why should a frontend hosting account and a private data account not automatically have the same network exposure requirements?

7. Audit Key Vault Diagnostic Settings

Key Vault is security-sensitive because it stores secrets, keys, or certificates.

Without diagnostic logs, it is harder to answer questions such as:

- Who accessed a secret?
- Which operation was performed?
- Did the operation succeed or fail?
- From which IP address or identity did the request come?
- When did access happen?

7.1. Check diagnostic settings

1. Open the Key Vault.
2. Go to **Monitoring**.
3. Open **Diagnostic settings**.
4. Check whether a diagnostic setting sends audit logs to the Day 5 Log Analytics workspace.

Record the result:

Resource	Diagnostic setting present?	Destination
Key Vault		

> **Note**
>
> When Key Vault audit logs are sent to Log Analytics using resource-specific tables, they commonly appear in the `AZKVAuditLogs` table.

> **Question**
>
> If Key Vault audit logs are not sent to Log Analytics, what kind of investigation becomes harder?

8. Audit App Service HTTPS Enforcement

The App Services should enforce HTTPS.

Check both backends:

- Public backend App Service.
- Internal backend App Service.

In the Azure portal:

1. Open the App Service.
2. Open the relevant configuration or TLS/SSL settings page.
3. Confirm that **HTTPS Only** is enabled.

Optional CLI check:

```
az webapp show `
  --resource-group <RESOURCE_GROUP_NAME> `
  --name <APP_SERVICE_NAME> `
  --query "{name:name, httpsOnly:httpsOnly}" `
  --output table
```

> **Note**
>
> If Terraform sets `https_only = true`, this should already be enabled. Still verify it, because audits compare the intended state with the real deployed state.

9. Audit Network Security Rules

NSGs were introduced earlier in the course. In this lab, review them from a governance perspective.

> **Note**
>
> Network Security Groups control traffic at subnet or network interface level. Storage account networking rules control access to the storage service endpoint. These are related, but they are not the same control.

Open the Network Security Groups in your resource group and review:

- Inbound rules.
- Outbound rules.
- Priority order.
- Source and destination.
- Protocol and port.

Record anything that looks unusual:

NSG / rule	Finding	Why it matters

> **Question**
>
> Is a deny-all outbound rule always a good idea? What legitimate traffic might an application still need?

10. Document Your Findings

Based on your audit, complete the checklist.

Mark each item as:

```
Pass
Fail
N/A
Needs review
```

Check	Status	Notes
Main resources have required tags		
Frontend storage exposure is understood and documented		
Private/data storage public network access is restricted or justified		
Storage accounts deny anonymous blob access unless explicitly required		
Key Vault sends audit logs to Log Analytics		
App Services enforce HTTPS only		
NSG rules are documented and justified		
Infrastructure encryption decision is documented for storage accounts		
Private endpoints are enabled where required by the target environment		

11. Identify Critical Versus Informational Findings

Classify your findings.

11.1. Critical or high-priority findings

Use this category for issues that could directly increase the chance of data exposure, loss of auditability, or unauthorized access.

1.

2.

3.

11.2. Informational or improvement findings

Use this category for items that are best practices, documentation gaps, or lower-risk improvements.

1.

2.

3.

> **Question**
> Which finding would you fix first if this were a production environment? Why?

12. Reflection Questions

Your answer:

1. Why is it dangerous to rely on manual audits for compliance checks?
2. Which finding was easiest to detect manually?
3. Which finding would be easy to miss without a checklist or policy?
4. How could you prevent a development configuration from accidentally being promoted to production?
5. Why is missing Key Vault diagnostic logging a security risk?
6. Which findings should become Azure Policy guardrails in the next lab?

13. Summary

In this lab, you manually audited the Day 5 deployment for governance and compliance findings.

You reviewed:

- Terraform environment settings.
- Resource tags.
- Storage account exposure.
- Key Vault diagnostic settings.
- App Service HTTPS enforcement.
- Network security rules.

You did not fix the findings yet. Instead, you documented them and classified their severity.

The key lesson is:

“A manual audit helps you understand the environment, but scalable governance requires repeatable controls such as policy, infrastructure-as-code, monitoring, and remediation.”

Next, you will manually fix selected findings to establish a governance baseline before enforcing it with Azure Policy.