

# THE 7 DEADLY SINS OF AZURE MISCONFIGURATION AND HOW TO FIX THEM

## OVERVIEW

In the Microsoft Cloud, where there are few perimeters and an almost unlimited arsenal of ephemeral cloud resources, there are many different ways in which an adversary can get direct access to your cloud environment if you make the wrong configuration change. Given the pace of innovation of the cloud and the speed at which many businesses are adopting Azure, it's far too easy to miss a step along the way and leave your business vulnerable to attack.

So, how do you know whether a misconfiguration is going to put you at risk? And how do you identify where *your* gaps actually exist? In this whitepaper, we'll walk through the 7 deadly sins of Azure misconfiguration and show you how to fix them.

Let's start by realizing what your responsibility is when it comes to your cloud security.

## SIN #1: ABDICATING RESPONSIBILITY OF CLOUD SECURITY

Simply put, cloud security in Azure is a shared responsibility. Depending on whether you are using IaaS, PaaS or SaaS, ultimately **you** are responsible for how clients access **your data**. In other words, the data is yours. And your responsibility. Microsoft does have a shared responsibility in your cloud environment. You can refer to the chart to the right for a breakdown of Microsoft's responsibilities and yours as the customer.

It's important to note that as you start leveraging more managed services from Microsoft like their managed database offerings for things like MySQL, PostgreSQL and Redis Cache that the lines blur between the application and the service. They will be backing up the infrastructure and data. However, it is still your responsibility to make sure that you have offsite / off-service backups of that data.

Responsibility	SaaS	PaaS	IaaS	On-prem
Data governance & rights management	Customer	Customer	Customer	Customer
Client endpoints	Customer	Customer	Customer	Customer
Account & access management	Customer	Customer	Customer	Customer
Identity & directory infrastructure	Shared	Shared	Customer	Customer
Application	Microsoft	Shared	Customer	Customer
Network controls	Microsoft	Shared	Customer	Customer
Operating system	Microsoft	Microsoft	Customer	Customer
Physical hosts	Microsoft	Microsoft	Microsoft	Customer
Physical network	Microsoft	Microsoft	Microsoft	Customer
Physical datacenter	Microsoft	Microsoft	Microsoft	Customer

Legend:  Microsoft  Customer

### CONSIDER AZURE BACKUP

One of the more interesting additional services you can add to many of your services like Azure Compute is the ability to automatically backup the cloud resource using Azure Backup. In many cases it's a simple checkbox. Use it. However, still make sure you conduct data backups away from the service.

## SIN #2: GIVING TOO MUCH ACCESS

### Control what users can access

By far the most common misconfiguration in Azure when it comes to user access control is giving people more permission to cloud resources than is required to do their job. Microsoft's Role Based Access Control (RBAC) provides *Fine-Grained Access Control* which can be applied to cloud resources hosted in Azure.

The way you control access to Azure resources is to create role assignments. This is a key concept to understand as it's how permissions are enforced. A role assignment consists of three elements: a *security principal*, the *role definition*, and a *scope*.

A *security principal* is an object that represents a person, a group or a service principal (application) that is requesting access to an Azure resource. Once you know who/what the subject is, you can assign them to a role.

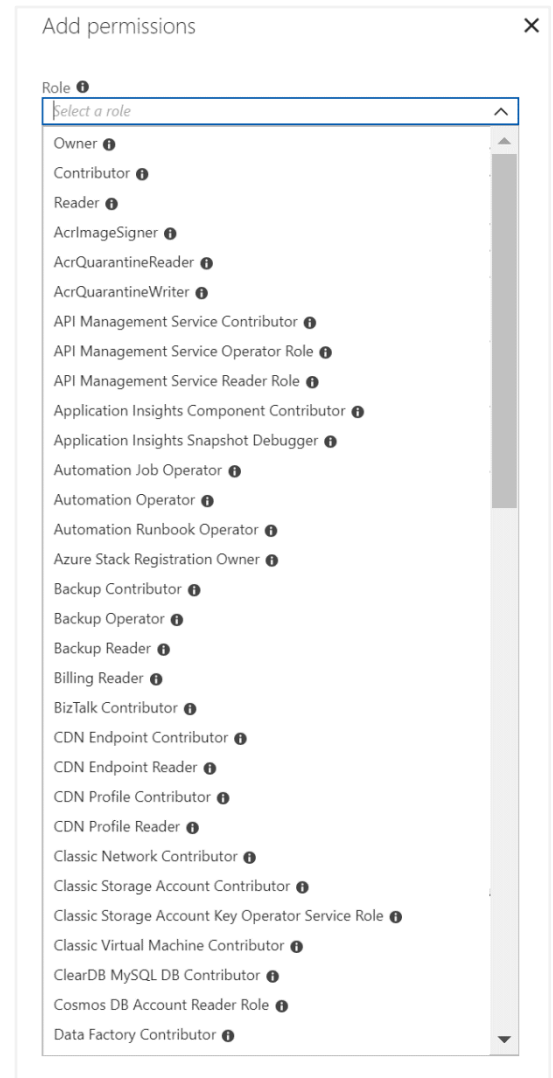
A *role definition* represents a collection of permissions. We usually just call them *roles*. A role definition lists the operations that can be performed, such as read, write and delete. There are four fundamental built-in roles you should be aware of:

1. [Owner](#) - Has full access to all resources including the right to delegate access to others.
2. [Contributor](#) - Can create and manage all types of Azure resources but can't grant access to others.
3. [Reader](#) - Can view existing Azure resources.
4. [User Access Administrator](#) - Lets you manage user access to Azure resources.

The rest of the built-in roles allow management of specific Azure resources. For example, the [Virtual Machine Contributor](#) role allows a user to create and manage virtual machines. If the built-in roles don't meet the specific needs of your organization, you can create your own custom roles.

Once roles are assigned, you can apply it to a *Scope*. A scope is the boundary that the access applies to. In Azure, you can apply a scope at multiple levels: subscription, resource group or resource. Scopes are structured in a parent-child relationship where every child will only ever have one parent. Therefore, children inherit permissions from the parent. For example:

- If you assign the *Reader* role to a group at the subscription scope, the members of that group can view every resource group and resource in that subscription.
- If you assign the *Contributor* role to an application (via a service principal) at the resource group scope, it can manage resources of all types in that resource group, but no other resource groups in that subscription.



## BE CAREFUL WITH CUSTOM SUBSCRIPTION OWNER ROLES

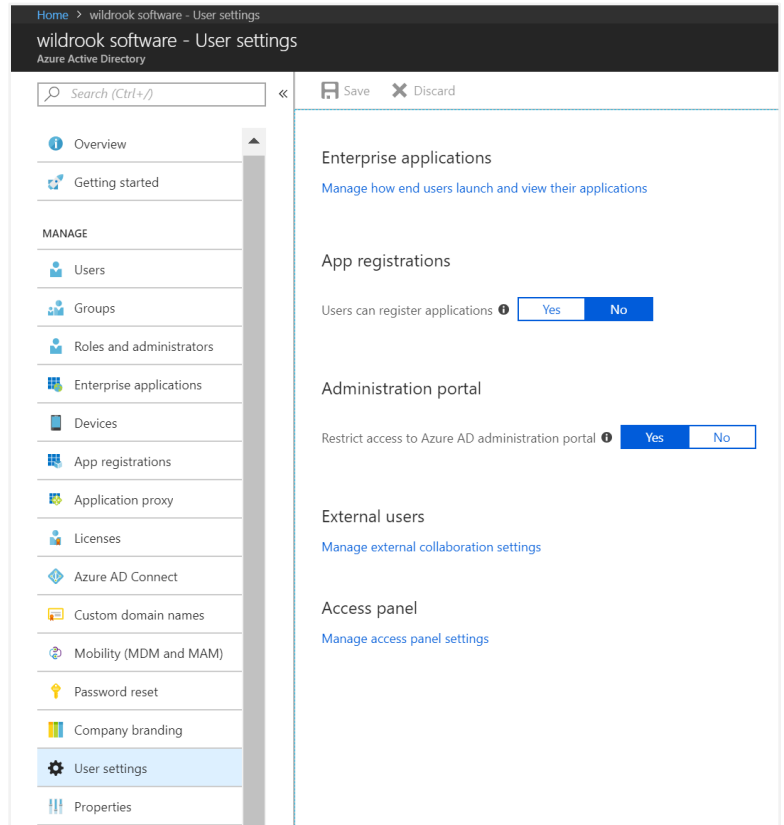
When possible, do not create custom roles with subscription ownership. It is recommended to use the principle of least privilege, assigning only needed privileges to the necessary resources or resource groups instead of allowing full administrative access to everything in a subscription.

## Restrict access to the Azure AD portal

The Azure AD administrative portal has sensitive data. You should restrict all non-administrators from accessing any Azure Active Directory data in the administration portal to avoid exposure.

To do this, follow these steps:

1. Login into the Azure Portal
2. Go to **Azure Active Directory**
3. Go to **Users settings**
4. Ensure that **Restrict access to Azure administration portal** is set to **Yes**.



## WARNING

While this action does prevent non-administrative users from being able to look at your Azure Active Directory data from the portal, it is still possible for them to programmatically access this information via PowerShell or the Azure API/CLI.

## Use Multi-Factor Authentication

Statistics show that more than 80% of the breaches that occur in the cloud do so through privileged accounts. Typically, this is due to the simple weaknesses in password security. It is far too easy to share, steal or guess a credential that might be used in a web application. Azure is no exception. While there are several strong countermeasures built into the Microsoft Cloud to battle credential abuse, a compromised account of an authorized administrator is still a primary target for threat actors.

To battle this, it should be required that all cloud administrator accounts have Multi-Factor authentication **enforced**. This includes users in roles like:

- Global Administrators
- Service Co-Administrators
- Subscription Owners
- Contributors

For more information on how to setup MFA in Azure, please [go here](#).

### ENFORCE MFA ON ADMINS

Users assigned the Global Administrator role in Azure AD tenants can enable multi-factor authentication for their Azure AD Global Admin accounts at no additional cost. For subscription administrators/co-administrators you can add MFA to Azure AD for just \$1.40 USD/month per admin.

Cost of strong authentication is no longer an excuse or barrier to enabling this security control. Require it now on all cloud administrators.

Also consider enabling MFA for your end users too. The same rationale for applying stronger assurances that the individual attempting to gain access is who they claim to be applies to these users. Combined with device trust, you can provide significant improvements in authenticating users to the Microsoft Cloud (both Azure and Office 365) without impeding their ability to get work done.

## Disable Guest Access

Azure AD includes the ability to support B2B collaboration, allowing you to invite people from outside your organization to be guest users in your cloud account. Until you have a business reason to do so, you should avoid this option. As guest accounts are not typically part of your employee on-boarding/off-boarding process they can potentially lay around unnoticed indefinitely leading to potential vulnerability.

Worse yet, if you do allow guest access, make sure you turn off their ability to invite other users to your cloud resources. This helps to maintain the 'Need to Know' and inadvertent access to your data.

To disable this capability, follow these steps:

1. Login into the Azure Portal
2. Go to **Azure Active Directory**
3. Go to **Users settings**
4. Go to **External collaboration settings**
5. Ensure that **Guests can invite** is set to **No**.

## SIN #3: FAILING TO USE BUILT-IN DATA SECURITY CONTROLS

Azure includes several key security controls that help to safeguard data at rest, and in transit. It is your responsibility to make sure that these are turned on. Using RBAC to secure the data stored in Azure is useful to ensure that only the necessary privileges are assigned to authorized users and applications. In addition, the security of the data itself can be enhanced by using features such as storage-side encryption and Azure Disk Encryption. As these are not always on by default, it is easy to miss this. Especially if you have cloud resources that have been around for some time.

Recently, Microsoft has started to make encryption the default on newly created cloud resources like storage accounts. However, it is important to check to make sure this encryption is turned on, as it is not guaranteed that older resources have it applied. Below are a few guidelines for how to protect your data in Azure.

### Ensure data is encrypted at rest

Storage service encryption protects your data at rest. Azure Storage encrypts your data as it's written in its datacenters, and automatically decrypts it for you as you access it. To make sure it's turned on, follow these steps:

1. Go to **Storage Accounts**
2. For each storage account, go to **Encryption** under **BLOB SERVICE**
3. Ensure that **Storage service encryption** is set to **Enabled**.

For more information on how encryption works for data at rest, [see this article](#).

#### WARNING

After enabling Storage Service Encryption, only new data will be encrypted immediately. Existing data in the storage account will be retroactively encrypted by a background encryption process that may take some time to complete.

#### SAFER DEFAULTS ARE COMING

Microsoft is rolling out a new data at rest strategy for Azure where storage accounts will automatically be encrypted and cannot be disabled. They are further enhancing this by allowing you as the customer to maintain your own keys if you so desire. You can learn more about customer-managed keys for data at rest using Azure KeyVault [here](#).

## Ensure data is encrypted in transit

You can enhance the security of your data by only allowing requests to storage accounts over a secure connection. For example, when calling a REST API to access your storage account you can enforce the need to use HTTPS. Any requests made over an insecure transport will be actively rejected. This includes Azure File Services. Scenarios that use SMB 2.1, SMB 3.0 without encryption and some versions of the Linux SMB client will fail when this option is enabled.

To enforce encryption in transit, follow these steps:

1. Go to **Storage Accounts**
2. For each storage account, go to **Configuration**
3. Ensure that **Secure transfer required** is set to **Enabled**.

### WARNING

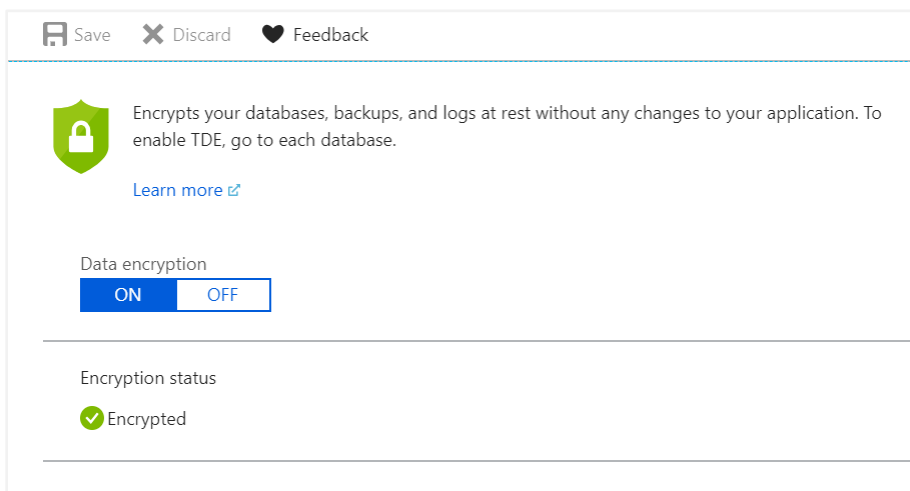
As Azure Storage does not support HTTPS for custom domain names, this feature is not applied to custom domains, even if the portal says it is Enabled. To learn more, please [read this article](#).

## Ensure SQL databases are encrypted


Azure SQL Database transparent data encryption (TDE) helps protect against the threat of malicious activity by performing real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application.

To verify that TDE is enabled, follow these steps:

1. Go to **SQL databases**
2. For each DB instance, under Settings go to **Transparent data encryption**
3. Ensure that **Data encryption** is set to **On**.



Save Discard Feedback

 Encrypts your databases, backups, and logs at rest without any changes to your application. To enable TDE, go to each database.

[Learn more](#)

Data encryption

ON  OFF

Encryption status

Encrypted

## Ensure “OS disks” and “data disks” are encrypted for VMs

Encrypting the operating system (OS) boot volume ensures that its entire content is fully unrecoverable without a key and thus protects the volume from unwarranted reads. This is important to ensure the integrity of the configuration of the machine as well as the privileges account credential stores that may exist on the system.

To verify that the OS disk is encrypted, follow these steps:

1. Go to **Virtual machines**
2. For each virtual machine, go to **Settings**
3. Click on **Disks**
4. Ensure that the **OS disk** has encryption set to **Enabled**.

Repeat these steps to protect your data for any and all data disks attached to your VMs.

### DISKS NOT ENCRYPTED?

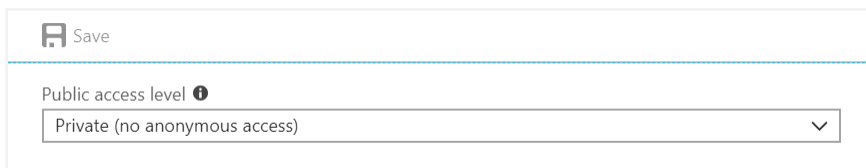
For more information on how to enable disk encryption if it is not enabled, [read this article](#).

## Ensure storage blob containers are not public

Azure Blob Storage allows you to grant anonymous read-only access to data in your storage account. It is highly recommended to not provide anonymous access to blob containers until and unless it is strongly desired. You should use shared access signature token for providing controlled and timed access to blob containers instead.

To check to see if you are allowing anonymous public access, follow these steps:

1. Go to **Storage Accounts**
2. For each storage account, go to **Containers** under **BLOB SERVICE**
3. For each container, click **Access Policy**
4. Ensure that **Public access level** is set to **Private (no anonymous access)**



The screenshot shows a 'Save' button at the top left. Below it, the 'Public access level' is displayed with a help icon. A dropdown menu is open, showing 'Private (no anonymous access)' as the selected option.



## Ensure storage keys are regenerated regularly

When you create a storage account, Azure generates two storage access keys which are used for authentication when the storage account is accessed. Rotating these keys periodically ensures that any inadvertent access or exposure to these keys could be undermined.

To check for any keys that have been rotated in the last 90 days, follow these steps:

1. Go to **Storage Accounts**
2. For each storage account, go to **Activity log**
3. Under the **Timespan** drop-down, select **Custom** and choose **Start time** and **End time** such that it ranges 90 days
4. Enter **RegenerateKey** in the **Search** text box
5. Click **Apply**

### NOTE

As the Activity Log only goes back 90 days when conducting a custom search, if you do not retrieve any records, then it's time to consider regenerating keys now.

## Ensure that Shared Access Signature (SAS) tokens expire in a timely manner

A Shared Access Signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a SAS token to clients who should not be trusted with your storage account keys but whom you wish to delegate access to certain storage account resources. By distributing a SAS token to these clients, you grant them access to a resource for a specific period of time.

This time should be set to a period as low as possible, preferably within an hour. And you should only allow this token over HTTPS by enforcing it under **Allowed protocols**.

The screenshot shows the configuration options for a Shared Access Signature (SAS) token. The settings are as follows:

- Allowed services:**  Blob,  File,  Queue,  Table
- Allowed resource types:**  Service,  Container,  Object
- Allowed permissions:**  Read,  Write,  Delete,  List,  Add,  Create,  Update,  Process
- Start and expiry date/time:**
  - Start: 2017-06-28 1:00:00 PM
  - End: 2017-06-28 2:00:00 PM
  - Timezone: (UTC-07:00) --- Current Timezone ---
- Allowed IP addresses:** 168.1.5.65
- Allowed protocols:**  HTTPS only,  HTTPS and HTTP
- Signing key:** key1

A blue button labeled "Generate SAS and connection string" is located at the bottom of the form.

### WANT TO LEARN MORE?

For more information on how to delegate access with Shared Access Signatures, please [read this article](#).

## SIN #4: FAILING TO MONITOR CRITICAL ACTIVITY IN AZURE

Azure Activity Log provides powerful insights into what is happening with your Azure subscription in terms of access and management of resources. This service keeps an audit trail of a range of activity, from Azure Resource Manager operational data to updates on Service Health events. Using the activity log, you can determine the 'what, who and when' any write operations (PUT, POST, DELETE) is taken on a resource in your subscription.

These audit logs can be integrated with other monitoring systems for advanced analysis and reporting. You can also stream the Activity Log to Event Hubs, which would allow for automated event driven responses to key events.

It's important that you get familiar with the Azure Activity Log. It holds a wealth of information that can help you monitor your cloud resources more effectively. There are several key categories of data you should be looking at, including:

- **Administrative** – This category contains the record of all create, update, delete, and action operations performed through Azure Resource Manager (ARM).
- **Service Health** – This category contains the record of any service health incidents that have occurred in Azure. Take special note for events for "Action Required" and "Security".
- **Alert** – This category contains the record of all activations of Azure alerts.
- **Autoscale** – This category contains the record of any events related to the operation of the autoscale engine based on any autoscale settings you have defined in your subscription.
- **Recommendation** – This category contains recommendation events from Azure Advisor.
- **Security** – This category contains the record of any alerts generated by Azure Security Center.

Here is the best way to get quick access to Activity Logs across all of Azure:

1. Go to **Monitor**
2. Select **Activity log** under **SHARED SERVICES**
3. Under the **Timespan** drop-down, select **Custom** and choose **Start time** and **End time** such that it ranges 90 days
4. Click **Apply**
5. Notice the returned audit records. For more details, click the event and select **JSON** to see the raw data of the event.

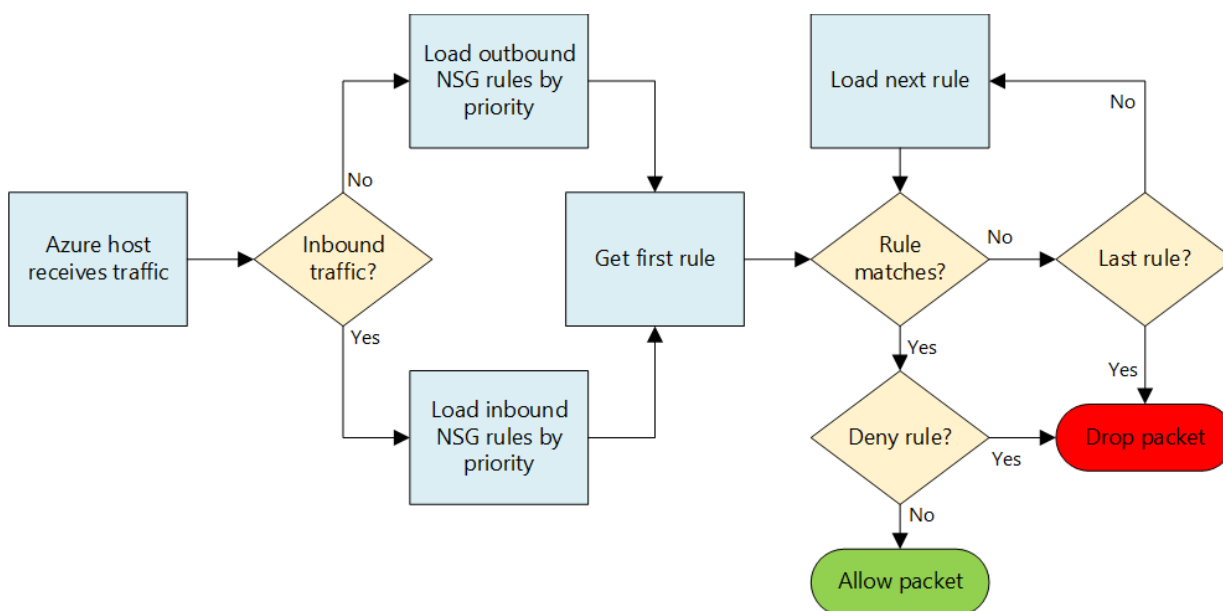
### GETTING TO KNOW AZURE ACTIVITY LOG

A great place to get a more in-depth understanding of Azure Activity Log can be [found here](#).

## SIN #5: FAILING TO USE NETWORK SECURITY GROUPS PROPERLY

Network Security Groups (NSG) provides segmentation with a Virtual Network (VNet) as well as provide full control over traffic that ingresses or egresses a cloud resource in a VNet. You can think of it as a basic firewall for your cloud resources. You can use NSGs to filter traffic by source and destination IP address, port and protocol.

Organizations often configure NSGs with too broad a permission scope, which can invite unwanted access to your Azure environment. This usually happens when there are conflicts between access rules set at the NIC level and subnet level of the VNet. Understand that NSGs are evaluated independently, and an "allow" rule must exist at both levels otherwise the traffic will not be authorized. The image below should help clarify the concept.



### Be very careful with Defaults

When creating a Network Security Group, even completely empty without any rules, there are some defaults that come with it, which you can visualize in ARM accessing the specific property of the NSG object in PowerShell:

```
Rules :
```

Type: Inbound							
Name	Priority	Action	Source Address Prefix	Source Port Range	Destination Address Prefix	Destination Port Range	Protocol
ALLOW VNET INBOUND	65000	Allow	VIRTUAL_NETWORK	*	VIRTUAL_NETWORK	*	*
ALLOW AZURE LOAD BALANCER INBOUND	65001	Allow	AZURE_LOADBALANCER	*	*	*	*
DENY ALL INBOUND	65500	Deny	*	*	*	*	*

Type: Outbound							
Name	Priority	Action	Source Address Prefix	Source Port Range	Destination Address Prefix	Destination Port Range	Protocol
ALLOW VNET OUTBOUND	65000	Allow	VIRTUAL_NETWORK	*	VIRTUAL_NETWORK	*	*
ALLOW INTERNET OUTBOUND	65001	Allow	*	*	INTERNET	*	*
DENY ALL OUTBOUND	65500	Deny	*	*	*	*	*

Notice in the input and output that there are three inbound and outbound rules. Rules are assigned a priority, and these default rules cannot be deleted. Instead, if you wish to override them you need to add new rules with a higher priority.

## Be careful on “Deny All” outbound Internet traffic

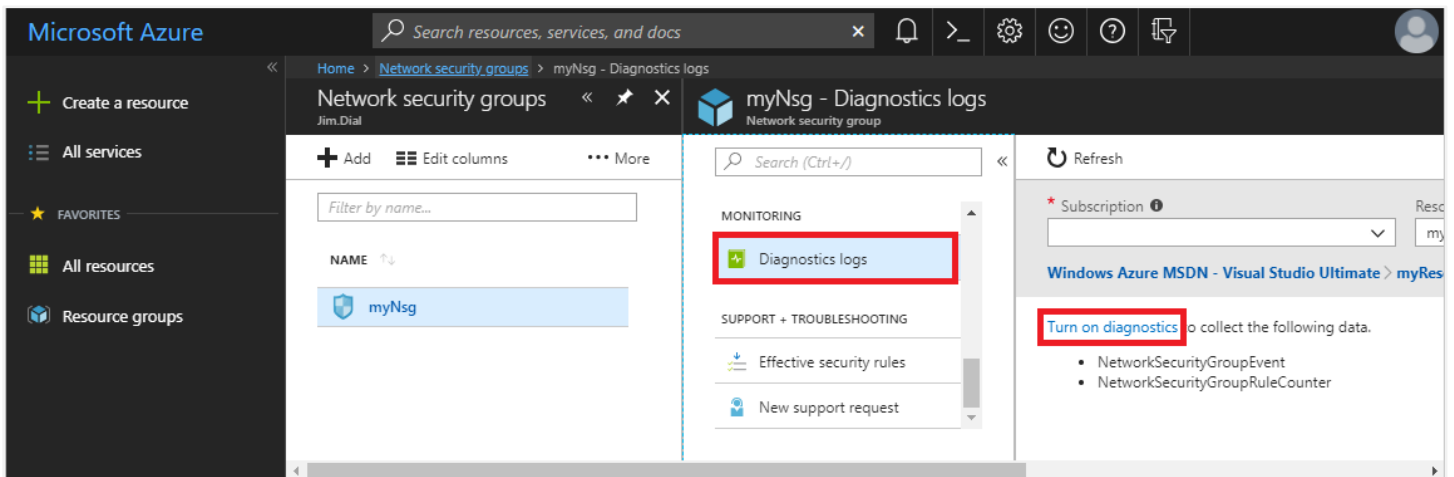
It is good network hygiene to deny all traffic outbound, helping to harden how the cloud resource(s) interact with other things. However, take into consideration how things like VM Extensions may stop functioning if you do that. As it has to reach out to an Azure storage account to write out a **.status** file in the VM’s storage account, blocking traffic may cause interruptions in service.

A better way would be to restrict connectivity while still allowing the required VM agent connectivity by adding NSG rules permitting Internet connectivity to only the Azure public IP address ranges for the region where the VM resides. You can learn more about how to manage NSG rules to work in this configuration [here](#).

## Enable NSG diagnostics and logs

When working with NSGs its important to enable diagnostics and logs to troubleshoot misconfigurations. Follow these steps to do that:

1. Go to **Monitor**
2. Select **All services**, then type *network security groups*. When **Network security groups** appear in the search results, select it.
3. Select the NSG you want to enable logging for.
4. Under **MONITORING**, select **Diagnostic logs**, and then select **Turn on diagnostics**.
5. Under **Diagnostic logs**, enter a name for the log, select a storage account to save to and then select **Save**.



### TIP

Consider using the **Activity Log Alerts** capability to send alerts when an NSG or NSG Security Rule is CREATED, UPDATED or DELETED. You want to be immediately alerted if the attack surface of your Azure environment changes. For more information on how to do this, [read this article](#).

## SIN #6: IGNORING EPHEMERAL RESOURCES

The dynamic nature of Azure's services means that cloud administrators and members in DevOps can easily deploy new infrastructure in an instant. With Infrastructure as Code (IaC) as a primary value chain for things like Azure Resource Manager (ARM) templates, it's a powerful way to leverage the cloud.

However, attackers with cloud access can also spin up cloud resources, use them for nefarious activities like data exfiltration, and remove it before the cloud administrators notice something is wrong. It's important that you stay vigilant and always know what resources are in use in Azure, who has created or destroyed such resources, and why.

You can use Azure Activity Logs and report on when such activity occurs with your cloud resources. Here is one way to do it:

1. Go to **Monitor**
2. Select **Activity log** under **SHARED SERVICES**
3. Under the **Timespan** drop-down, select **Custom** and choose **Start time** and **End time** such that it ranges 90 days
4. Under **Event Category** select **Administrative**
5. Under the Operation drop-down type *create* in the text box and then select the types of create events you want to filter for.
6. Click **Apply**
7. Notice the returned audit records. For more details, click the event and select **JSON** to see the raw data of the event.
8. Repeat the steps above for a *delete* filter to search for deleted resources.

Another option is to monitor your resource usage, adjust spending limits and limit the resource count quota authorized for allocation. In this way, you can stay on top of such activities and prevent extreme resource cost overages.

### WARNING

Using spending limits as a countermeasure to battle ephemeral resources in a subscription for production services is usually a bad idea. Once a spending limit is reached your services get disabled, and virtual machines will be deallocated.

### TIP

Consider streaming CREATE and DELETE activity logs for resources to EventHub, and then use serverless compute with Azure Functions to take action when events are created. This way you can have custom alerts/notifications send almost anywhere, like a webhook to a Microsoft Teams or Slack channel, or an SMS to the on-call cloud administrator.

## SIN #7: ALLOWING CONFIGURATION DRIFT

One of the most powerful things about Azure is also one of its biggest detriments. It is far too easy to make a configuration change in the Azure portal or through the Azure CLI / PowerShell. As time goes on, administrators and application owners may need to make modifications to their cloud resources to continuously improve the product they provide to their users. As those modifications and changes happen in response to business needs, the configuration of the applications and infrastructure changes, in many cases without other administrators knowing. These changes might be benign, or they might take the systems out of a hardened state. This is known as “configuration drift”.

Depending on the severity of the drift, there could be significant risk to the business. It is therefore important that cloud resource configurations that change are properly accounted for and documented.

There are a few ways to tame configuration drift. These include:

1. **Baseline Configuration** – Establish an approved baseline configuration of your cloud environment. You can leverage Microsoft’s [“Export ARM template”](#) functionality in Azure to help with this.
2. **Automation** – Leverage standardization and automation of provisioning of your accepted baseline configuration through patterns making it easier to maintain consistent configurations across all environments. Things like [ARM templates](#) and [Azure Automation](#) help here.
3. **Change Orchestration** – Watch the Azure Activity Log for CHANGE events to cloud infrastructure and ensure there is properly documented *change requests* explaining why it was done, and under whose authority. Then update your baseline configuration to match these new production settings.
4. **Continuously assess configuration state** – Use tools that can more closely evaluate your approved baseline and alert you when configuration drift occurs.

### TIP

Tools like AuditWolf ([www.auditwolf.com](http://www.auditwolf.com)) are designed to help you continuously assess your Azure configurations, provide an audit trail of configuration changes and alert you when such changes impact your cloud security posture. Consider using the CIS Benchmark policies to assess your environment regularly.

## NEXT STEPS

Azure misconfiguration is a common issue. While this whitepaper has covered 7 of the most common areas of concern, there is a lot more guidance on how to properly configure and manage Azure cloud resources out there. Below are some helpful reference links you should check out.

- [Azure Security Best Practices and Patterns](#)
- [Azure Data Security and Encryption Best Practices](#)
- [Security Management in Azure](#)
- [Azure Logging and Auditing](#)

## ABOUT AUDITWOLF

What if you don't want to spend the time to manually check your security posture through the Azure Portal? **AuditWolf has your back.** It is your **cloud threat protection platform** for Azure that continuously monitors your cloud environment looking for *risky configurations, vulnerable hosts, poor data security practices* and *risky privileged accounts*. In just a few clicks AuditWolf can help you identify and prioritize risk in your Azure environment and provide you guidance on exactly how to fix it. Interested in learning more? Click the button below to get started!

## Free Cloud Risk Assessment

Let's look at your Azure environment and see how many of the **7 deadly sins** affect you!

TRY IT FREE