***Firewall:-***

Azure Firewall is a cloud-native, managed network security service in Azure that provides firewall functionality to control and secure network traffic. It’s used to centrally govern and log all network traffic flows across Azure VNets. Here’s a breakdown of how it works, along with key configurations and setup instructions.

**1. Understanding Azure Firewall Basics**

Azure Firewall operates at Layer 4 and Layer 7 of the OSI model:

* **Layer 4 (Transport Layer)**: Allows or denies traffic based on IP addresses and port numbers (TCP/UDP).
* **Layer 7 (Application Layer)**: Filters traffic based on URLs, FQDNs (fully qualified domain names), and protocols such as HTTP/HTTPS.

It enables you to create rules to control inbound and outbound traffic to/from subnets and restrict access to certain applications and services.

**2. Key Features of Azure Firewall**

* **Network Rule Collection**: Controls access based on IP addresses, protocols (TCP, UDP, ICMP), and port numbers. Ideal for non-HTTP/S traffic.
* **Application Rule Collection**: Allows or denies access based on FQDNs. Useful for managing web-based applications (HTTP, HTTPS).
* **Threat Intelligence**: Detects and blocks traffic from known malicious IP addresses and domains using Microsoft Threat Intelligence.
* **Network Address Translation (NAT)**: Configures inbound rules to allow external access to resources within VNets.
* **Logging and Analytics**: Logs all traffic and integrates with Azure Monitor for analytics.

***Parent Policy:***

* If your firewall policy is inherited from another policy (a parent policy), this section shows the source policy. In multi-layered policies, parent policies provide inherited rules and configurations.

***Rule Collections****:*

* **Rule collections** are sets of rules that define the firewall’s behaviour. These are divided into three main types: DNAT, Network, and Application rules.

***Network Rules****:*

* **Network rules** control traffic based on IP addresses and protocols. They define rules for allowing or denying access based on source and destination IP, port, and protocol.

***Application Rules***:

* **Application rules** are used to filter traffic based on fully qualified domain names (FQDNs). These rules are especially useful for web traffic filtering, as they focus on application-level inspection.

**Destination Network Address Translation (DNAT)**

1. **Purpose**:
   * DNAT is primarily used to translate the destination IP address of incoming traffic. It allows external users to access services hosted on internal resources (like VMs) using a public IP address.
2. **Functionality**:
   * It rewrites the destination IP address of the incoming packets to direct them to the correct private IP address of a VM based on defined rules.
   * Commonly used with Azure Firewall or other network appliances.
3. **Usage Scenario**:
   * Used when you want to expose services running on a VM or service to the internet, allowing external traffic to reach specific internal resources (like a web server or application).
   * Example: Redirecting traffic from the firewall's public IP (e.g., 20.188.232.197) to an internal VM's private IP (e.g., 10.0.1.4) on a specific port.
4. **Protocol**:
   * Operates primarily at Layer 4 (Transport Layer), dealing with TCP/UDP traffic.
5. **Configuration**:
   * DNAT rules are configured in Azure Firewall, specifying the source, destination IP, and ports.

**Setting Up DNAT Rules for VM1 and VM2**

Here’s how to set up separate DNAT rules for VM1 and VM2 in Azure Firewall:

**Step 1: Access Azure Firewall**

1. **Go to Azure Portal**: Log in to your Azure portal.
2. **Select Your Firewall**: Navigate to your Azure Firewall instance.

**Step 2: Create a DNAT Rule Collection**

1. **Navigate to Rules**:
   * Click on **Rules** in the left menu, then select **DNAT rules**.
2. **Add a DNAT Rule Collection**:
   * Click on **Add DNAT Rule Collection**.
   * **Name**: Common\_VM\_DNAT\_Collection
   * **Priority**: Choose a priority number (lower number means higher priority).
   * **Action**: Set to Allow.

**Step 3: Create DNAT Rules for Each VM**

1. **Create DNAT Rule for VM1**:
   * Click on **Add DNAT Rule** within the collection you just created.
   * **Name**: DNAT\_VM1
   * **Source Type**: Select IP Address.
   * **Source**: Set to \* (allows traffic from any IP) or specify allowed IPs for tighter control.
   * **Destination IP**: Set to the firewall’s public IP (e.g., 20.188.232.197).
   * **Destination Port**: Enter the port on which you want to access VM1 (e.g., 80 for HTTP, 443 for HTTPS).
   * **Translated IP**: Enter the private IP of VM1 (e.g., 10.0.1.4).
   * **Translated Port**: The same as the destination port (e.g., 80 or 443).
2. **Create DNAT Rule for VM2**:
   * Click on **Add DNAT Rule** again.
   * **Name**: DNAT\_VM2
   * **Source Type**: Select IP Address.
   * **Source**: Set to \* or specify allowed IPs.
   * **Destination IP**: Set to the firewall’s public IP (e.g., 20.188.232.197).
   * **Destination Port**: Enter the port on which you want to access VM2 (e.g., 8080 for an application).
   * **Translated IP**: Enter the private IP of VM2 (e.g., 10.0.1.5).
   * **Translated Port**: The same as the destination port (e.g., 8080).
3. **Save the Rules**: Click **Save** to create the rules.

**Example Configuration**

* **Public IP of Firewall**: 20.188.232.197
* **VM1 Private IP**: 10.0.1.4
  + **DNAT Rule for VM1**:
    - Source: \*
    - Destination IP: 20.188.232.197
    - Destination Port: 80
    - Translated IP: 10.0.1.4
    - Translated Port: 80
* **VM2 Private IP**: 10.0.1.5
  + **DNAT Rule for VM2**:
    - Source: \*
    - Destination IP: 20.188.232.197
    - Destination Port: 8080
    - Translated IP: 10.0.1.5
    - Translated Port: 8080

**Step 4: Testing**

1. **Accessing VM1**:
   * Open a web browser and go to http://20.188.232.197 (assuming VM1 is running a web server on port 80). This should route the request to 10.0.1.4.
2. **Accessing VM2**:
   * Open a web browser and go to http://20.188.232.197:8080 to access VM2, routing to 10.0.1.5.

**Summary**

* You can create multiple DNAT rules in a single DNAT rule collection to route traffic to different VMs.
* Each rule translates incoming traffic on the firewall’s public IP to the respective VM’s private IP based on the specified ports.
* Make sure to verify that your Azure Firewall has appropriate **Network Rules** and **Application Rules** to allow the intended traffic.

### User Defined Routes (UDR)

1. **Purpose**:
   * UDR is used to override Azure's default routing for network traffic within a virtual network or to direct traffic to specific resources based on user-defined rules.
2. **Functionality**:
   * UDR allows you to specify a static route for outbound traffic, determining how traffic is routed within your virtual network and to/from the internet.
   * You can direct traffic to specific network appliances or services, such as routing traffic through a firewall.
3. **Usage Scenario**:
   * Used to customize the routing behaviour for specific subnets or network traffic, often in scenarios involving multiple virtual networks or complex routing needs.
   * Example: Creating a UDR to route all outbound traffic from a subnet to an Azure Firewall IP address for inspection before reaching the internet.
4. **Protocol**:
   * Operates at Layer 3 (Network Layer), dealing with IP routing.
5. **Configuration**:
   * UDR is configured in the Azure portal under the route tables associated with specific subnets, allowing you to define routes based on destination IP addresses and next-hop types (e.g., Virtual Appliance, Internet).

Great! Since your DNAT rules are configured and working for inbound traffic, you’re already set up to handle incoming requests routed through the firewall to your VMs. If you’re able to make outbound requests (like curl http://example.com) from your VMs to the internet successfully, this means your setup is functional for outbound traffic as well.

**Do You Still Need UDR?**

Here’s how to evaluate the need for UDRs in your setup:

1. **Outbound Traffic Routing to the Firewall (Optional)**:
   * If you specifically want all outbound traffic from your VMs to pass through the Azure Firewall for security, logging, or monitoring purposes, then you would still add **User Defined Routes (UDRs)** for the outbound path.
   * This ensures that even responses or new outbound connections from your VMs go through the firewall, which may be a compliance or security requirement.
2. **If Outbound Traffic Already Works Without UDRs**:
   * Since outbound traffic seems to be working without UDR, this could be because Azure’s default routes are allowing the outbound traffic directly from the VMs to the internet.
   * If you don’t have a specific requirement to inspect or filter outbound traffic via the firewall, then UDRs may not be necessary.
3. **When to Use Both DNAT and UDR**:
   * If **strict control** over both inbound and outbound traffic is needed, configure both DNAT (for inbound) and UDR (for outbound to firewall).
   * This configuration ensures that all traffic, whether inbound or outbound, is directed through the firewall for consistency.

**Summary**

* **If your setup requires** all outbound traffic to be inspected by the firewall: Add UDRs on the VM subnets to route outbound traffic through the firewall.
* **If outbound traffic inspection is not required** and everything is working as needed: UDRs may not be necessary.

Let me know if you want further details on setting up UDRs, or if you need specific routing requirements.

**To set up User Defined Routes (UDR**) for your two VMs (VM1 and VM2) so that all their outbound traffic routes through the Azure Firewall, follow these steps:

**1. Create a Route Table**

1. Go to the **Azure Portal**.
2. Search for **Route tables** and click on **+ create** to create a new route table.
3. Set the following details:
   * **Subscription**: Select your subscription.
   * **Resource Group**: Choose the same resource group where your VMs and firewall are located.
   * **Name**: Give a name, such as VMs\_RouteTable.
   * **Region**: Select the same region as your VMs.
4. Click on **Review + create**, and then **Create**.

**2. Add a Route to the Route Table**

1. Once the route table is created, navigate to it and select **Routes** under **Settings**.
2. Click on **+ Add** to add a new route:
   * **Route Name**: Provide a name, like OutboundTrafficToFirewall.
   * **Address Prefix Destination**: Enter 0.0.0.0/0 (this directs all outbound traffic).
   * **Next Hop Type**: Select **Virtual appliance**.
   * **Next Hop Address**: Enter the **private IP address** of the Azure Firewall (e.g., 10.0.4.196).
3. Click **OK** to save the route.

**3. Associate the Route Table with Each VM Subnet**

1. In the route table, go to **Subnets** under **Settings**.
2. Click on **+ Associate** to associate the route table with each subnet:
   * **Virtual Network**: Select the virtual network where your VMs are hosted.
   * **Subnet**: Select the subnet of VM1 (e.g., Subnet1 for frontend or Subnet2 for backend).
3. Repeat this step to associate the route table with the subnet of VM2.

This will direct all outbound traffic from both VMs to the Azure Firewall for inspection, filtering, or logging.

***The "Action: Deny. Reason: No rule matched"*** response means that the Azure Firewall is blocking the outbound request because there isn't an outbound rule explicitly allowing this traffic. By default, Azure Firewall denies all traffic unless there's a rule allowing it.

To resolve this, you need to add an **Application Rule** or **Network Rule** to the Azure Firewall to allow outbound HTTP/HTTPS traffic to the internet. Here’s how to set it up:

**Steps to Add an Outbound Rule in Azure Firewall**

1. **Go to Azure Firewall**:
   * In the Azure Portal, navigate to your **Azure Firewall** instance.
2. **Configure Application Rule (Recommended for HTTP/HTTPS traffic)**:
   * Under **Settings**, select **Rules** and go to the **Application Rule Collection** tab.
   * Click on **+ Add application rule collection**.

Fill in the following details:

* + **Name**: Give it a name, such as Allow-HTTP-HTTPS.
  + **Priority**: Choose a priority (lower numbers mean higher priority).
  + **Action**: Set to **Allow**.
  + **Rule Name**: Give the rule a name, such as AllowWebTraffic.
  + **Source Type**: Set to **IP Address**.
  + **Source IP addresses**: Enter the IP ranges for VM1 and VM2’s subnets (e.g., 10.0.1.0/24 and 10.0.2.0/24).
  + **Protocol**: Select **HTTP, HTTPS**.
  + **Target FQDNs**: Enter \* if you want to allow all websites, or specify specific domains if you want to limit access (e.g., www.example.com).

1. **Save the Rule Collection**:
   * Once you’ve filled in the details, click **Add** to save the rule.
2. **Test Connectivity**:
   * Return to your VM and run curl http://www.example.com again to confirm that outbound HTTP traffic is now allowed.

This configuration allows your VMs to access internet sites via HTTP and HTTPS while still routing traffic through the Azure Firewall for outbound inspection. Let me know if you encounter any additional issues!

Using

Dig command can check where the traffic is directing

Example : - dig [www.example.com](http://www.example.com)

Or   
we can log in to virtual machine and check the access log in var file and verify that it is working or not

***SNAT IP Prefixes***:-

Snat is an Azure feature that allows you to define custom outbound IP address prefixes for network traffic leaving your Azure Firewall. Instead of the default behaviour where Azure chooses from a pool of IPs, SNAT IP Prefixes give you control to specify the IP addresses used for Source Network Address Translation (SNAT) in your outbound traffic, offering benefits like more predictable IP usage and simplified firewall configurations on receiving networks.

***Example Scenario of SNAT IP Prefixes:-***

Suppose you have an Azure Firewall in a virtual network (VNet) and two web application VMs (VM1 and VM2) behind it. You want outbound traffic from these VMs to use specific IP addresses when reaching external services.

1. **Setup**:
   * You create a **Public IP Prefix** in Azure with a range of public IP addresses, say 20.30.40.0/28, which provides 16 IPs.
   * Assign this Public IP Prefix to the Azure Firewall using the SNAT IP Prefixes feature.
2. **Outcome**:
   * When VM1 or VM2 sends outbound requests (like accessing an external API or web service), Azure Firewall translates their private IPs to one of the IPs from the specified 20.30.40.0/28 prefix.
   * This means external systems always see the same set of IPs from this prefix, rather than random Azure-assigned IPs.
3. **Benefits**:
   * **Predictable IP Addresses**: External systems that require IP whitelisting (like certain APIs or databases) can be configured to allow only traffic from your specified IP range.
   * **Compliance and Control**: If you’re subject to regulations or corporate policies requiring specific outbound IPs, this feature makes compliance easier to achieve.

**How to Configure SNAT IP Prefixes**

1. **Create a Public IP Prefix** in your Azure subscription, defining the range you need for outbound traffic.
2. **Assign the Public IP Prefix** to your Azure Firewall under the SNAT IP Prefixes configuration.
3. **Update Firewall Rules** as needed to allow outbound traffic from your VMs.

**Testing**

After setting up SNAT IP Prefixes, you can verify that outbound traffic from your VMs is using the specified IP range by checking the IP address observed on external services or websites.

This configuration provides consistency in IP addresses used for outbound traffic and simplifies scenarios where predictable IP ranges are needed.