Policy Objects

Objects Overview

Policy objects are resources used to define the match criteria referenced inside a policy rule. Traffic entering a gateway instance is then evaluated against this match criteria. Rules are the individual sequences inside the policy ruleset attached to a gateway or a set of gateway instances front-ended by load balancer. When traffic enters a gateway instance, each rule in the ruleset is evaluated against the incoming traffic in a strict order until a match is found, or the end of the ruleset is reached. For more information, see Rules.

What are these Objects for?

Need for Policy Objects

Objects play a crucial role in defining the type of traffic a particular rule is looking to match against and then take a specified action on. If traffic matches the object definitions referenced inside a particular Rule (AND logic for match), the matched traffic can be allowed or denied. If the traffic is allowed, additional advanced security profiles can be applied for further inspection. Each type of object helps define match criteria at a different layer of the OSI model.

Types of Policy Objects

- **Address Objects**: Address objects are used to define match criteria that is mapped to Layer 3 IP addresses. These objects are referenced inside a policy rule. Address objects can be defined using explicit IP addresses/CIDRs, FQDNs, or cloud-native resources discovered by the Multicloud Defense Controller through periodic asset discovery / real-time event tracking.
- **Service Objects**: Service objects are used to define Layer 4 match criteria that is referenced inside a policy rule. They are also used to define how a gateway instance will process an incoming traffic flow. This is referred to as the connection type.

  For example, if the Service objects are defined with a connection type of either Forwarding or Forward Proxy. When the connection type is set to Forwarding inside a Service object, the incoming traffic flow is passed through the gateway instance. No proxying and/or decryption occurs. In this case, you can use the Service object to define a set of destination ports and the associated protocol incoming traffic will be evaluated against when a specific policy rule is processed.

When the connection type is set to Forward Proxy inside a Service object, the incoming traffic flow is proxied through the gateway instance. Decryption will occur depending on the proxy type. In this case, you can use the Service object to define the proxy type as well as the destination port or set of ports the gateway instance will be listening on for packets sent from the client. The gateway instance will specifically listen for a HTTP Request packet or a TLS client Hello. Once it receives this packet, the gateway instance extracts the host information and uses it to establish a separate connection to the external host.

- **FQDN Match Objects**: FQDN Match Objects are used to define a set of FQDNs for explicit whitelisting or blacklisting. The gateway instance extracts the host information from a HTTP request or TLS client hello and uses it to match against the FQDNs listed in the FQDN Match Object. These FQDN Match Objects are evaluated against incoming traffic at Layer 5 and are particularly useful for matching against HTTP or HTTPS traffic, where host information is visible in the request packet.

Usage of Policy Objects

Once a policy object is defined, it is referenced inside a policy rule. It is important to note that although each of these objects can be referenced inside a policy rule, only Src/Dest Address Objects and Service Objects are mandatory objects when defining a rule. The FQDN Match Object is an optional parameter.

Matching traffic based on src/dest address objects, service objects and or FQDN match objects invokes the first two stages of the data path pipeline (L4 FW/FQDN Match) used to inspect traffic within each gateway instance. This is important to note because these stages are the first points of traffic inspection. The incoming traffic flow may be dropped at one of these stages or it may be allowed to pass through and be inspected at stages further along the pipeline correlating to the advanced security profiles referenced in the matched rule.

Static Objects

Static objects specifies unchanging IP addresses, subnets, or specific firewall rules to provide predictable and stable configurations which can be important for compliance and security purposes. In a cloud environment, this allows you to create and share objects that maintain the same IP address or FQDN within a hybrid environment.

If you choose to delete a shared object, Multicloud Defense deletes it only from its system. The object continues to exist within Security Cloud Control.

Dynamic Objects

In contrast, dynamic objects do not have to specify an IP address at all. Dynamic objects are adaptable configurations that automatically adjust to varying conditions or environments. They allow firewalls to respond to real-time events without requiring manual intervention.

You can also **tag** resources and use them as objects to create a more fine-tuned ruleset within your policy. This level of fleibiltity within a cloud environment allows the system to adjust for yoou based on real-time data and can result on reduced maintenance.

Sharing Objects with Security Cloud Control

In an environment where you may have cloud-based managers such as AWS or GCP interacting with on-premises datacenters, it is crucial to be able to share objects within policies to protect your environment. Shared objects make it easy to maintain policies because you can modify an object in one place and that change affects all other policies that use that object. Without shared objects, you would need to modify all the policies individually that require the same change.

Multicloud Defense shows you a combined or "flattened" view of the elements of the object in the details pane. Notice that in the details pane, the network elements are flattened into a simple list and not directly associated with a named object.

Note that sharing objects is only supported when you deploy an access control policy that allows traffic from your cloud-based datacenter. Ensure that your policy includes, or excludes, instances or attributes from your third-party datacenter.

When you share objects with Security Cloud Control they are automatically translated into **network objects**. This does not affect the original state of the object in Multicloud Defense.

To configure object sharing between Multicloud Defense and Security Cloud Control you **must** create a connector in Security Cloud Control and attach the connector to an applicable policy to enable this feature and then import objects to see them in the Multicloud Defense Controller. See About the Multicloud Defense Connector for more information.

If you happen to share dynamic objects there is the option to preserve the original values of the object by creating an override value. An object override allows you to override the value of a shared network object on specific devices. See Object Overrides for more information.

Note Objects cannot be shared with Cloud-delivered Firewall Management Center.

Resource Tagging in Objects

In a traditional policy configuration, traffic is matched against objects and profiles; these constructs often include stagnant IP addresses that must be manually and routinely updated. By including resource tags in your objects, you can avoid mismatches with incorrect or stale IP addresses by matching traffic against a fluid resource tag that is defined by its type, name, or value and not just an IP address or address range. Since tags are not dependent on the cloud service provider you are using, you can easily group and share the same tags or have multiple tags used in policies without incident.

As of now, Multicloud Defense supports resource tagging with the following inventory types:

- Subnets
- VPCs/VNets
- Instances

Once you add a tag to an inventory item and attach a value you can add it directly to your policy or even build a policy around the tags you have created. See How to Tag Object Resources, on page 3 for more information.

How to Tag Object Resources

Object resources are found in the **Inventory** page of the Multicloud Defense dashboard. Tagging these resources can improve your firewall policies with enhanced efficiency, scalability, and visibility of network security management.

Be aware that you can only tag the following object resources:

- Subnets
- VPCs/VNets
- Instances

Before you begin

This procedure utilizes tags and resources that are already created. If your tenant does not already have inventory resources or tags available to use, create them first. See Inventory and Tags for more information. Note that you may have to enable asset discovery to allow Multicloud Defense access to assets and objects in your cloud service povider accounts if you have not already.

Procedure

Step 1 In the Security Cloud Control platform menu, choose **Products** > **Multicloud Defense** .

Step 2 Navigate to the **Inventory** tab and select the object resource of your choice from the list of resources above that support tagging.

Step 3 Select at least one rule from the table of rules and then click **Add Tags**. You can select multiple rules if you want to add the same tag to more than one at a time.

Step 4 In the Tags window, expand the drop-down menu of the **Key** column and select one of the available options.

Missing a Tag? If you do not see the Key that you want to use in this situation, click your cursor into the drop-down menu field and manually type the desired name of the key and click the "**+Add**" button to the immediate right of the text field. This tag is only applicable to the resource you manually tag it to and is not available in the drop-dow menu after saving the object.

Step 5 Manually enter a value in the **Value** column.

Step 6 Confirm the values of the tag and the selected resources. If the selections are satisfactory, click **Save**. Optionally, click **Cancel** to remove the configuration.

What to do next

Add a tagged object resource to a policy.

Add a Tagged Object to a Policy

Use the procedure below to add a tagged resource to your policy:

Before you begin

You must have the following already completed before you continue with this procedure:

- An onboarded cloud service provider with asset discovery enabled. See Enable Asset Discovery and Inventory for more information.
- At least one VPC or VNet compatible with the cloud service provider that is already onboarded.

- At least one object. See Inventory for more information.
- At least one tagged object resource. See How to Tag Object Resources, on page 3 for more information.

Note this procedure does not support OCI cloud service providers.

Procedure

Step 1 In the Security Cloud Control platform menu, choose **Products** > **Multicloud Defense** .

Step 2 Navigate to **Policies** > **Security Policies** > **Addresses**.

Step 3 Check the box of an existing Src/Dest policy from the table or click **Create** to create a new policy and select **Src/Dest** as the policy type.

Step 4 Enter a unique **Name**. Note that the name cannot contain any spaces.

Step 5 (Optional) Enter a **Description** for the policy. This can be helpful to differentiate between other policies that might have a similar naming convention.

Step 6 Expand the **Type** drop-down menu and select **User Defined Tag**.

Step 7 Expand the **CSP** drop-down menu and select a cloud serice provider that is laready onboarded to your tenant.

Step 8 Expand the **Region** drop-down menu and select the region associated with the cloud service provider.

Step 9 Expand the **VPC** drop-down menu and select the VPC or VNet you want associated with the policy.

Step 10 Expand the **Subnet** drop-down meu and select the subnet you want assigned to the policy.

Step 11 (Azure only) Expand the **Resource Group** drop-down menu and select the group you want associated with the policy.

Step 12 The **Resource Level** refers to the granularity at which security rules can be applied to control traffic between source and destination resources. Resource-level policies enable administrators to define security rules for specific resources, such as virtual machines, subnets, or specific IP addresses, rather than applying broad rules across network segments. Expand the **Resource Level** drop-down menu and select one of the following options:

- VPC/VNet
- Instance
- Subnet
- Load Blancer
- Security Group
- (Azure only) Application Security

Step 13 Expand the **Resource Tag** drop-down menu and select the name of the tag you want to apply. This tag references all the object resources you added to it.

Step 14 Expand the **Value** drop-down menu and select the autopopulated options that are avaiable. These options are tied directly to the resource tag itself and cannot be used interchangbly across other tags unless specifically configured that way.

Step 15 Confirm the matching expression at the very bottom of the window. If this is correct, click **Save** to save and apply the policy to your cloud service provider. Alternatively, click **Cancel** to delete the temporary configuration and not deploy.

Address Objects

Address Objects

An **Address object** represents a set of one or more IPs, CIDRs or FQDNs for use as a **source** or **destination** in a **security policy rule set rule**, or as a **target backend address** in a **reverse proxy service object**, depending on how it is defined. The address object can be configured statically using traditional constructs or dynamically using cloud constructs.

An address object represents a set of one or more IPs, CIDRs or FQDNs within a **Source**, **Destination**, or **Reverse Proxy Target** field within a security policy rule or rule set. You can also define it as a target backend address within a reverse proxy service object. This section focuses on source and destination objects.

These are the general guidelines to follow when you define address objects:

- Initially create address objects using static IP addresses and CIDRs, and confirm proper matching.
- When replacing address objects, use address objects that use dynamic cloud resource (tags, labels) and confirm proper matching.
- Ensure that the private address object includes both RFC 1918 and RFC 6598.
- Ensure that the Internet address object does not include RFC 1918 and RFC 6598.

As of Version 24.04 and later, you can now configure an address object to **exclude** specific IP addresses or an IP address range.

Src/Dest

These objects are used to define match criteria that maps explicitly to IP addresses or CIDRs. The objects are referenced inside a policy rule and are evaluated against traffic entering a gateway instance when a policy rule is processed.

Source and destination address objects are useful when IP Addresses and CIDRs are explicitly needed to match application traffic entering a gateway instance. These objects are referenced inside the source and destination fields of a policy rule definition. The type of address object used to populate each of these fields depends on the traffic flow, application type, and use-case.

Source or Destination Address Objects

A source or destination address object specifies a source or destination for a rule inside a security policy rule set. It is used by the rule to match traffic based on its source or destination IP address. The different types of address objects are defined as follows:

IP/CIDR/FQDN (Static) Address Objects

An IP/CIDR/FQDN address object is configured as a set of IP addresses, CIDR blocks or FQDNs. Examples of IP/CIDR address objects include:

- Destination IPs for DNS servers.
- Destination IPs for SMTP Relay Servers.
- Destination IPs for NTP servers.

- Source IPs or subnets for application workloads.

FQDN address objects define an explicit set of FQDNs for allowing or blocking IPs based on DNS resolution. When you define an FQDN inside an FQDN address object and reference it in a policy rule, the gateway instance performs a DNS resolution to retrieve the corresponding IP address(es) to match incoming traffic. By default, caching is not enabled. In this case, the DNS resolution is done every 60 seconds, and the gateway instance uses the retrieved resolution for 60 seconds. If the FQDNs specified inside the FQDN address object are resolving to a large set of IP addresses (i.e. more than 400 each), then caching can be enabled. In this case, the DNS resolution interval can be specified, along with the cache size and cache TTL.

FQDN address objects are useful to match on application traffic that is either UDP based (ex. NTP) or TCP traffic for which host information does not exist in the request packet (ex. SMTP). In either case, it is recommended to use an FQDN address object to match on this kind of application traffic instead of manually defining a list of IP addresses for all appropriate NTP servers or SMTP servers, for example, your internal workloads are required to connect to.

Dynamic Cloud Constructs

Cloud-Native address objects are dynamic cloud resources discovered by the Multicloud Defense Controller through either periodic inventory collection (API-based) or real-time event tracking (GCP Pub/Sub integration). These resources can be individual resources such as VPCs/VNETs, Instance IDs, security groups, Subnet IDs or a set of resources referenced through user-defined tags. The multicloud defense controller uses a combination of real-time event tracking and targeted API calls to dynamically populate the IP addresses associated with the cloud resource. Therefore, any subsequent changes made to a cloud-native resource with be automatically reflected inside the address object referencing this resource.

Note Using cloud-native constructs to define source or destination address objects allows you to create a truly dynamic cloud policy across both single and multi-cloud environments. As cloud resources are added, deleted, or changed within a cloud environment, the address objects are dynamically updated to reflect these changes, making sure your security posture is automatically updated across all applications and functions in your environment.

User-Defined Tags in VNet and VPC Environments

Tags map the IP addresses or CIDR for a cloud resource defined with a set of tags to an address object. In GCP, labels are key-value pairs that are often used to categorize resources dedicated to different environments (i.e., development, staging, production, etc.). Inside a source or destination address object, user-defined tags can be used to reference resources including instances, VPCs/VNETs, subnets, and security groups. Most commonly, organizations use tags to categorize instances.

Tag based policy rules are a very powerful component of dynamic cloud policies. Granular policy rules can be defined for groups of instances with specific tags. With these policy rules in place, anytime a new instance is deployed with the appropriate tags, it automatically inherits the desired security policy defined for the category of instances it belongs to. This is because the Multicloud Defense Controller does not only discover a new instance has been deployed, but also the tags that have been assigned to that instance. It will then dynamically update the source or destination address object referencing this instance-based tags with the new instance’s IP address. If an instance is deployed with the incorrect tags or no tags, it will not be allowed to communicate to any other resources because the appropriate policy rule is not matched against.

In VNets and VPCs, tags map the CIDR associated with the VPC to an address object CIDR. Provides a contextual way of creating a rule that matches any instance deployed within a VPC or VNET. It can use the name of a discovered VPC or VNET to define match criteria instead of having to manually figure out what CIDR is associated with a particular VPC or VNET. Any changes to the VPC or VNET will be dynamically updated in the policy rule with no intervention. If a VPC or VNET is removed and a new VPC/VNET is created in its place, the rule will no longer apply even if reusing the CIDR.

Instance ID

Instance IDs map the IP addresses associated with an instance to a list of IP addresses inside an address object. This provides a contextual way of creating a policy rule for a specific instance without manually figuring out how the instance is configured. The policy rule reflects any changes to the instance or its removal. Note that the policy rule cannot apply to any other instance, even if the instance is deleted and replaced with a new instance with the same configuration.

Security Group

Security Groups map the IP addresses of network interfaces associated with a security group to a list of IP addresses inside an address object. Any interface related changes, such as fields that are added or removed to the security group, are dynamically reflected in the list of IP addresses inside the address object. This provides an organization with the ability to align existing security groups with the advanced security capabilities of the gateway data path pipeline.

Subnet IDs

Subnet IDs map the CIDR associated with a subnet to an address object CIDR. This provides a contextual way of creating a policy rule for all resources associated with a specific subnet ID without manually figuring out how the subnet is configured. A VPC or VNET is typically divided into multiple subnets and resources deployed within these subnets may serve different purposes. For example, instances in one subnet may require a specific set of advanced security profiles or may have a different traffic flow requirement. To simplify the process of creating different security rules for each subnet, Multicloud Defense gives you the capability to define a policy rule using the subnet's name as match criteria. Therefore, each subnet can have a unique policy rule, with unique security profiles. Any changes to the subnet and any instance deployed within the subnet is dynamically reflected in the policy rule.

Geo IP

These objects are used to allow or block traffic that is coming from or going to IP addresses based on their geographic location (country). Multicloud Defense integrates with the MaxMind GeoIP2 Database for maintaining a list of updated GeoIPs.

To review a full list of country names and codes, or IP address to GeoIP country codes, go to the GeoNames website.

Group

A group address object is configured as a set of sour or destination address objects. A group provides flexibility by defining individual address objects and then grouping them together, simplifying the number of rules necessary to match traffic based on the members of the group. The group inherits the set of IPs, CIDRs, or FQDNs from the members of the group, whether the members are static, dynamic, or a combination of the two.

Source or Destination Address Object Parameters

Type	Mode: Dynamic or Static	Parameter	Required or Optional	Notes
IP/CIDR/FQDN	Static	Value	Required	The total number of FQDNs per Address Object is limited to 200 where each FQDN can resolve to at most 400 IPs. The Multicloud Defense Gateway will perform DNS resolution every 60 seconds, regardless of the DNS record TTL.
VPC/VNet ID	Dynamic	CSP Account	Required	
		Region	Required	
		Resource Group	Optional	Azure Only
		VPC/VNet ID	Required	
Security Group	Dynamic	CSP Account	Required	
		Region	Required	
		VPC/VNet ID	Required	
		Resource Group	Optional	Azure Only
		Security Group ID	Required	
Application Security Group	Dynamic	CSP Account	Required	Azure Only
		Region	Required	
		Resource Group	Required	
		Application Security Group	Required	
Instance ID	Dynamic	CSP Account	Required	
		Region	Required	
		VPC/VNet ID	Required	
		Resource Group	Optional	Optional
		Instance ID	Required	

Type	Mode: Dynamic or Static	Parameter	Required or Optional	Notes
Subnet ID	Dynamic	CSP Account	Required	
		Region	Required	
		VPC/VNet ID	Required	
		Resource Group	Optional	Azure Only
		Subnet ID	Required	
User Defined Tag	Dynamic	CSP Account	Optional	
		Region	Optional	
		VPC/VNet ID	Optional	
		Resource Group	Optional	Azure Only
		Resource/Tag/Value	Required	List of Resources and Tag Key-Value Pairs.Resources can be Instance, VPC/VNet, Subnet, Load Balancer, Security Group, Security Group (Azure).
Geo IP		Value	Required	
Group		Address	Required	

Reverse Proxy Target Address Object

A reverse proxy target address object is specified as a backend target address in a reverse proxy service object. It is used by the service object to establish a backend connection to an application. The application can be the address of one or more application load balancers or instances in the form of IPs or FQDNs. The different types of reverse proxy target address objects are defined as follows:

Static IP/FQDN Address Object

An IP/FQDN address object is configured as a set of IP addresses or FQDNs. When more than one IP or FQDN is configured, the gateway handles the addresses without priority amongst the configured fields when setting up a backend connection. When an FQDN is configured, the gateway resolves the FQDN with DNS to determine the IP address to use when setting up a backend connection.

Dynamic Applications Address Object

An applications address object is configured as an individual application load balancer cloud resource determined by its applications tag. The configuration dynamically populates a set of IPs or FQDNs represented by the cloud resources, obtained from the cloud account using the Multicloud Defense real-time inventory discovery. Any changes to the cloud resources will be automatically reflected in the address object. When the configuration results in more than one IP or FQDN, the gateway handles the fields with no priority amongst the set when setting up a backend connection. When the configuration result is an FQDN, the gateway will resolve the FQDN with the DNS to determine the IP address to use when setting up a backend connection.

Reverse Proxy Target Address Object Parameters

Type	Mode: Dynamic or Static	Parameter	Required or Optional	Notes
IP/FQDN	Static	Value	Required	
Applications	Dynamic	CSP Account	Required	
		Region	Required	
		VPC/VNet ID	Required	
		Resource Group	Optional	Azure Only
		Tag/Value	Required	Single Tag Key-Value pair

System Objects

Multicloud Defense provides a list of pre-defined address objects to simplify policy creation. All system objects cannot be deleted or modified. Users can choose to clone system objects if modification is needed.

Name	Description
Any	This represents the entire IPv4 address space.
any-private-rfc- 1918	This represents all IPv4 private address as defined in RFC-1918.
Internet	This represents the entire IPv4 public address space, minus the private IPv4 addresses (RFC1918).

Create a Source/Destination Address Object

For information on what this object is, see Source or Destination Address Object Parameters, on page 9. Use the following procedure to create a src/dst address object in Multicloud Defense:

Procedure

Step 1 Go to **Policies** > **Security Policies** > **Addresses**.

Step 2 Click **Create**.

Step 3 Select **Src/Dest**.

Step 4 Enter a unique **Name** to identify the address object.

Step 5 (Optional) Enter a description for the object. This may provide context to help differentiate the object from other objects.

Step 6 Select the **Object Type**. For information on object types and what they are, see Address Objects, on page 6. Select one of the following types:

- IP/CIDR/FQDN
- VPC/VNet ID
- Security Group
- Application ID (Azure only)
- Instance ID
- Subnet ID
- User-Defined Tag
- Geo IP
- Service End Point (Cloud Service IP)
- Group

Note
If you select **Group**, you can include a specific IP address or a range of IP addresses to either include or exclude.

Step 7 Depending on which type you selected in step 6, enter the following paramters:

- **Value** - Enter a valid IP, CIDR, or FQDN IP address.
- **CSP Account** - Use the drop-down menu to select a cloud service provider account that has already connected to the controller.
- **Region** - Select the region your cloud service provider is located in.
- **VPC** - Use the drop-down menu to select the VPC or VNet. Note that options available may change depending on the cloud service provider account your choose.
- **Subnet** - Use the drop-down menu to select the subnet that applies to your VPC or VNet.
- (Azure only) **Resource Group** - Use the drop-down menu to select the resource group that is compatible with your selections.
  - **Resource Level** - Use the drop-down menu to select a value.
  - **Resource Tag** - Use the drop-down menu to select a keyword as the resource tag.
  - **Value** - Enter a valid value for the resource group. Note that this is different from the Value entry expected for IP/CIDR/FQDN objects.
- **Geo IP** - Use the drop-down menu to select a specific IP that is associated with the gelocation of your choice.
- **X-Forwarded-For Match Enabled** - Check this box to allow the gateway to match against XFF HTTP header fields.
- **Address** - Select an existing object. This selection determines the group of addresses that

- **Include Addresses** - This option is only applicable if you select "Group" as the type in step 6. Enter a specific IP address or a range of IP addresses to include. You can also use `any` to include all valid addresses.

- **Exclude Addresses** - This option is only applicable if you select "Group" as the type in step 6. Enter a specific IP address or a range of IP addresses to exclude. You can also use `any` to include all valid addresses. Note that there is no validation from the Multicloud Defense Controller for address exclusion.

Step 8 (Optional) Include a **Matching Expression**. This represent the set of conditions which must be matched for the object to execute.

Step 9 Click **Save** when complete.

Create a Reverse Proxy Target Address Object

For more information on what this object is, see Reverse Proxy Target Address Object Parameters, on page 11. Use the following procedure to create a reverse proxy target address object in Multicloud Defense:

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **Addresses**.

Step 2 Click **Create**.

Step 3 Select **Reverse Proxy Target**.

Step 4 Enter a unique **Name** to identify the address object.

Step 5 (Optional) Enter a description for the object. This may provide context to help differentiate the object from other objects.

Step 6 Select the **Object Type**. For information on object types and what they are, see Address Objects, on page 6. Select one of the following types:

- IP/CIDR/FQDN

- Applications

Step 7 Depending on which type you selected in step 6, enter the following paramters:

- **Value** - Enter a valid IP, CIDR, or FQDN IP address.

- **CSP Account** - Use the drop-down menu to select a cloud service provider account that has already connected to the controller.

- **Region** - Select the region your cloud service provider is located in.

- **VPC** - Use the drop-down menu to select the VPC or VNet. Note that options available may change depending on the cloud service provider account your choose.

- **Subnet** - Use the drop-down menu to select the subnet that applies to your VPC or VNet.

- (Azure only) **Resource Group** - Use the drop-down menu to select the resource group that is compatible with your selections.

Step 8 Use the drop-down menus to select both an existing **Applications Tag** and its **Value** for this object.

Step 9 Click **Save** when complete.

Edit Address Objects

If you need to modify a parameter that cannot be modified, you will need to Clone the address object and then change the parameters as desired.

Use the following steps to edit an address object. Note that not all parameters can be edited.

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **Addresses**.

Step 2 Check the box next to the address object you would like to **Edit**.

Step 3 Click **Edit**.

Step 4 Modify the parameters as desired.

Step 5 Click **Save** when complete.

Clone Address Objects

If the desire is to use the clone in place of the original, you will need to replace all associations of the original with the clone. The associations will be in a set of one or more security policy rule set rules or reverse proxy service objects. The associations can be seen by viewing the Address Object Details.

Use the following steps to clone an existing address object:

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **Addresses**.

Step 2 Check the box next to the address object you would like to **Clone**.

Step 3 Click **Clone**.

Step 4 Specify and modify the parameters as desired.

Step 5 Click **Save** when complete.

Delete Address Object

If an address object is actively used in a policy rule set or a reverse proxy service object, it will have one or more associations, and you will be unable to delete the address object. In order to delete an address object, you must first remove all associations, then the address object can be deleted. The associations can be seen by viewing the Address Object Details.

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **Addresses**.

Step 2 Check the box next to the address object you would like to **Delete**.

Step 3 Click **Delete**.

Step 4 Click **Save** to confirm the delete.

View Details

You can view the address object **Details** by clicking the **Name** of an object from the **Policies** > **Security Policies** > **Addresses** page. The **Details** will display the IPs, CDIRs and FQDNs populated based on its type and configuration. It will also display the associations with policy rule sets and any object services.

Considerations and Best Practices

The Gateway can be configured to maintain an IP cache through Terraform. For more information, see Gateway FQDN IP Cache Settings.

FQDN Objects

FQDN Match Object

A Fully Qualified Domain Name (FQDN) Match Object evaluates the Server Name Indication (SNI) associated with TLS-encrypted traffic or the Host header for unencrypted HTTP traffic. It uses the results of the evaluation for rule matching. If the traffic matches all match objects (Address, FQDN, Service) associated with a rule, then the rule is used to process the traffic. To evaluate the FQDN, traffic must be TLS encrypted and contain an SNI in an unencrypted TLS Hello header or be unencrypted HTTP and contain a Host header. The FQDN can be evaluated for traffic that is processed by either a **Forwarding** or **Forward Proxy** rule. The set of FQDNs in the profile is specified as strings representing the full domain or as strings represented by a Perl Compatible Regular Expression (PCRE).

Note The FQDN match object is organized as a table containing user-specified rows (FQDNs).

The rows do not contain log-related actions to perform. This is because FQDN match object is a first-level matching criteria. When you have a clear list of FDQNs that you want to allow, you can use FQDN match objects. After a rule match, if you have categories that you want to allow based on criteria, use FQDN filtering. For more information, see Fully Qualified Domain Name Filter Profile.

The limits for each FQDN match object are as follows:

- Maximum user-specified rows: 254 (Standalone or Group of Standalones)
- Maximum FQDNs per row: 60
- Maximum FQDN character length: 255

When specifying a multilevel domain (for example, `www.example.com`), it's important to escape the `.` character (for example,`www\.example\.com`), otherwise it treats it as a wildcard for any single character.

Syntax

When defining an FQDN inside an FQDN match object, use PCRE formatting. The commonly used special characters are:

- Period (.) to match any character except for line terminators (\n).
- Asterisk (*) to match the previous pattern between zero and unlimited number of times, as many times as possible.
- Backward slash (\) to escape a special character. To specify a period character in an FQDN, please use \. Because the period alone is a special character. For example, in PCRE format, www.google.com would be written as www\.google\.com
- Question mark (?) to match the previous pattern between zero and one times, as many times as possible. This character is useful when you would like to make a specific pattern in an expression, optional.
- {1} to match the previous pattern one time exactly.

When using PCRE, it is important to test that the expression matches the desired FQDN strictly and does not match any undesired domains. The resources listed below can be used to gain a further understanding of PCRE syntax as well as test expressions to make sure they match the desired FQDN:

- PCRE Regex Cheatsheet
- PCRE Regex Testing Engine

For example, if we have a requirement to match on any subdomain of google.com as well as the domain itself, we can create an FQDN match object which includes the following FQDN definition **(.*\.)?google\.com**. When you read this from left to right, the expression says match any character any number of times (.*), followed by a period (\.) and make matching on this pattern optional (?). Then find google.com exactly (google\.com).

Here are the guidelines for defining FQDN match objccts:

- For all HTTPS and Websocket traffic, create a set of rules that use FQDN match objects. Each rule name will help distinguish how traffic is classified and processed by the gateway.

- Create separate rows in your FQDN match for domains that are being decrypted and domains that are not being decrypted (Decryption exception).
- Consider creating separate rules that use different FQDN match objects to decrypt and not decrypt.
- Ensure the PCRE domain expressions match desired domains and not undesired domains.

Standalone vs. Group

An FQDN Match Object can be specified as type Standalone or Group.

A FQDN Match Standalone Object contains FQDNs. The object will be applied directly to a set of one or more Policy Ruleset Rules or associated with a FQDN Match Group Object.

A FQDN Match Group Object contains an ordered list of Standalone FQDN Objects that can be defined for different purposes and combined together into a Group Object. The Group Object can be applied directly to a set of one or more Policy Ruleset Rules. Each team can create and manage specific Standalone Profiles. These Standalone Profiles can be combined together into a Group Profile to create hierarchies or different combinations based on use case. An example combination could be a global FQDN list that would apply to everything, a CSP-specific list that would apply to each different CSP, and an application-specific list that would apply to each different application.

Create Standalone FQDN Match Object

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **FQDN**.

Step 2 Click **Create**.

Step 3 Provide a Profile Name and Description.

Step 4 Specify the Type as Standalone.

Step 5 Click **Add** to create a new row.

Step 6 Specify individual FQDNs (e.g., www.twitter.com,.*.google.com).

a) Each FQDN is specified as a PCRE (Perl Compatible Regular Expression).
b) Consider escaping the . character, or else it will be treated as a single character wildcard.

Step 7 (Optional) Specify Decryption Exception for any FQDNs where decryption is not desired or possible. Possible reasons for considering Decryption Exception include:

- The desire to not inspect encrypted traffic (financial services, defense, healthcare, etc.).
- SSO authentication traffic where decryption is not possible.
- NTLM traffic that cannot be proxied.

Step 8 Click **Save**.

Create Group FQDN Match Object

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **FQDN**.

Step 2 Click **Create**.

Step 3 Provide a Profile Name and Description.

Step 4 Specify the Type as Group.

Step 5 Select an initial Standalone Profile (at least one Standalone Profile is required).

Step 6 Specify additional Standalone Profiles.

Step 7 Click **Add FQDN Profile** to create a new row.

Step 8 Select a Standalone Profile.

Step 9 Click **Save** when completed.

Associate the Object

To access and associate objects to policy rules, see Rules.

Block an FQDN Match Object

After you define an FQDN object, you can perform actions such as block the FQDN object in a ruleset. Blocking of an FQDN match object is applicable for egress traffic (Forwarding or Forward Proxy service objects). To block an FQDN match object, you will need match the object in a reference rule.

Note Version 24.08 supports 6-tuple matching with FQDN. This means if you opt to have the first matching rule configured to **block** a FQDN match, two events are logged instead of one: the first event logged for the L4 Firewall is "**Allow**" and the second event logged for the FQDN object match is "**Deny**".

Procedure

Step 1 Navigate to **Manage** > **Security Policies** > **Rule Sets**.

Step 2 Perform the steps outlined in Add or Edit a Forward Proxy Rule in a Rule Set.

Step 3 In the **Action** dropdown list, select **Deny Log**. This action will automatically drop the connection and deny the request.

Step 4 Click **Save** after completion of the outlined steps.

Service Objects

Service objects are used to define layer 4 match criteria (destination ports, protocol) within a policy rule. They are also used to define the connection type for an incoming traffic flow. For an Egress or East-West Gateway, the connection type is either set to Forwarding or Forward Proxy.

Reverse Proxy Service Object (Ingress)

Ingress service objects are used in the Ingress/Reverse proxy rules. The object defines a listener port that the Multicloud Defense gateway listens for the traffic it receives and forwards to the target/backend address. Listener port can be configured with a decryption profile that has a TLS certificate configured. When the traffic hits the listener port, Multicloud Defense Gateway returns the TLS certificate configured. consider the following configurable options:

- An SNI can be configured on this port. This enables a single listener port (e.g 443) to be proxied to multiple backend targets based on the SNI.
- L7 DoS (L7 Denial of Service) can be configured on the service to set rate limits for an URI and/or HTTP method.
- Target defines the backend address object and port to forward the traffic. The proxied traffic can be forwarded as HTTP, HTTPS, TCP, or TLS.

Use the following procedure to create and add a reverse proxy service object:

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **Services**.

Step 2 Click **Create**.

Step 3 Click **Reverse Proxy**.

Step 4 Provide a **Name** and **Description**.

Step 5 Configure proxy parameters as defined below:

Option	Description
Decryption Profile	Assign a decryption profile, which also includes the server certificate, to be used for the proxy service.
Dst Port	Assign a destination port. For most web-based services, the destination port will be 443. This is the port Multicloud Defense Gateway listens on for the incoming traffic.
Protocol	TCP is the default.
SNI	Enter the list of SNIs.
L7 DoS	Enter the Layer 7 DoS profile to assign to this proxy service.
Target Backend Port	Enter the Target/Backend application port number.

Option	Description
Protocol	Select the backend protocol.
Address	Select a backend IP address. The IP address in most cases will be the frontend IP of an internal load balancer.

Note
If the proxy service is required to run on multiple ports, you can add more entries. However all the ports serve the same certificate and are proxied to the same backend destination address object.

Forward Proxy Service Object (Egress / East-West)

Forward Proxy services are specifically used for HTTP based traffic. The object defines a listener port that the Multicloud Defense Gateway listens for the traffic it receives and forwards to the address/host that's available in the TLS SNI extension header or HTTP Host Header.

Note We recommend using this for egress/East-West traffic.

If the connection type is set to Forward Proxy, the traffic flow is proxied through the gateway at various layers depending on the proxy type. The session from the client is terminated on the gateway instance and a new session is established from the gateway instance to the destination. The gateway instance behaves as a mediator in the middle. The gateway instance listens for the HTTP host header or the TLS Hello packet. Once it receives the packet, it extracts the domain and connects to the host using the specified protocol and destination port. For encrypted traffic, a self-signed certificate is required to decrypt, inspect and re-encrypt traffic.

Operating in forward-proxy mode at the TLS layer requires a gateway instance to present a self-signed certificate to the client initiating the connection request. Self-signed certificate body is imported into the Multicloud Defense Controller. The associated private key can be imported to the Multicloud Defense Controller in the following ways:

- Import the private key.
- Store in AWS Secrets Manager and provide the secret name.
- Store in AWS KMS and provide the cipher text contents.
- Store in GCP Secrets Manager and provide the secret name.
- Store in Azure KeyVault and Secret and provide the keyvault and secret name.

Use the following procedure to create and add a forward proxy service.

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **Services**.

Step 2 Click **Create**.

Step 3 Click **Forward Proxy**.

Step 4 Provide a name and description.

Step 5 Optionally, select the Application IDs to match.

Step 6 Configure proxy parameters as defined below.

Option	Description
Decryption Profile	Assign a decryption profile, which includes the certificate to be used. Multicloud Defense impersonates the external certificate by signing it with the certificate provided in this profile. The root certificate is assumed to be installed on all the client application instances.
Dst Port	Assign a destination port. For most web-based services, the destination port will be 443.
Protocol	HTTP or HTTPS.

Note

- Multicloud Defense listens on the **Dst Port** and waits for the HTTP Host header or TLS SNI Header packet. Once Multicloud Defense receives this packet, it connects to the host using the protocol. If the protocol is HTTPS, the received certificate data from the external host is signed by the certificate in the decryption profile and sent to the client. The root certificate **must** be installed on the client app instances to avoid a certificate error.
- For a given destination port, there can be only one decryption profile (root CA certificate) association in a policy rule set across all service objects.
- During a forward proxy session, Multicloud Defense Gateway performs a DNS lookup on the destination with DNS request timeout of 30 seconds and cache age-out of TTL seconds.

Forwarding Service Object (Egress / East-West)

Forwarding service objects are used in the forwarding rules. The traffic that matches this type of rule/service is not proxied, and is forwarded as-is. This means there is no deep packet inspection and no Application ID on *encrypted* traffic.

Note We **strongly** recommend using this for East-West traffic.

If the connection type is set to Forwarding, the TCP session is passed through the gateway instance and terminated at the destination. No decryption is performed in this case. Forwarding is typically used when no decryption is required and matching on L3, L4, and L5 header information to allow or deny traffic is sufficient. Forwarding is also useful for traffic flows that are latency sensitive.

Although advanced security profiles such as Network Intrusion (IPS/IDS) can be turned on for rules that are Forwarding traffic, TLS decryption in conjunction with IPS is highly recommended for maximum protection against all malicious activity. If TLS decryption is not used, then an advanced security engine like IPS relies on heuristic scanning against a known database, which may result in false positives.

Use the following procedure to create and add a forwarding service object:

Procedure

Step 1 Navigate to **Policies** > **Security Policies** > **Services**.

Step 2 Click **Create**.

Step 3 Click **Forwarding**.

Step 4 Provide a name and description.

Step 5 Multicloud Defense supports source NAT on a per service level. For traffic that requires source IP preservation (e.g. East-West traffic), disable SNAT.

For Egress traffic, SNAT **must** always be enabled.

Step 6 Configure port parameters as defined below.

Option	description
Dst Port	Assign a destination port or a range of destination ports as `start-end`.
Protocol	TCP, UDP, ICMP

Note
In a forwarding policy, deep packet inspection operations **only** occur on non-encrypted traffic.